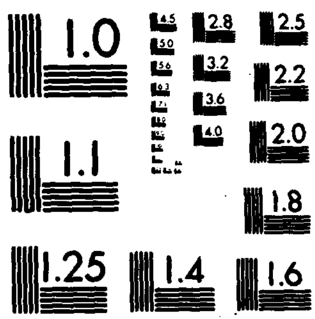
AD-A085 709 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 12/1
ROBUSTNESS STUDIES OF OUTPUT PREDICTIVE DEAD-BEAT CONTROL FOR W--ETC(U)
DEC 79 E H KIRKWOOD
UNCLASSIFIED AFIT/GA/EE/80-1 NL

1 of 2




MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

14

AFIT/GA/EE/80-1

11 Dec 79

9 Master's thesis,

6

ROBUSTNESS STUDIES
OF
OUTPUT PREDICTIVE DEAD-BEAT CONTROL
FOR
WING FLUTTER APPLICATION.

[redacted]

AFIT/GA/EE/80-1

10 Earl H. Kirkwood, Jr

[redacted]

A

12 136

Approved for public realease; distribution unlimited

012225 Glm

AFIT/GA/EE/80-1

# ROBUSTNESS STUDIES

# OF

# OUTPUT PREDICTIVE DEAD-BEAT CONTROL

# WING FLUTTER APPLICATION

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

in Partial Fulfillment of the

Requirements for the Degree of

Master of Science


by

Earl Kirkwood Jr.

2nd Lt. USAF

Graduate Astronautical Engineering


December 1979

# Preface

The intent of this report is to examine the robustness characteristics of a newly developed digital control law. This study will hopefully give insight to the use of the Output Predictive Dead-Beat Control law.

I wish to express my appreciation to my advisor, Professor J. Gary Reid for his guidance, assistance and patience throughout the year. His motivation and suggestions helped make this report as complete as possible.

Finally, I would like to thank my typist, Cathy Motsch, for her excellent work and suggestions.

## Contents

## List of Figures

| Figure | | Page |
|---|---|---|

| Figure | | Page |
|---|---|---|

## List of Tables

## Abstract

A new discrete control law ~~(Ref 3)~~ is implemented and examined. Robustness to noise and model errors of the control law as sample rate varies is analyzed. This analysis is conducted while controlling several different very lightly damped, single input/single output systems which are representative of the flutter dynamics of the B-52E wing . ~~(Ref 6).~~

The control law performance is different at separate sample rates. A distinct range of sample rates are found to have a better response to noise than other sample rates. Another range is found to be more robust when there exists an error in the models used to calculate the closed loop control law. When these ranges of sample rates intersect, the robustness characteristics at those sample rates is found to be good with respect to both noise and model mismatch.

As theoretically predicted, ~~in (Ref 3)~~ a relationship between condition number of the system Hankel matrix and robustness seems to exist. Hence, these simulated results appear to validate the theoretical results on robustness predicted by Reid, but on the other hand, these simulated results indicate that the total analysis of "robustness" is a very complex issue and cannot, at this point, be totally predicted by such a parameter as simple as the condition number of the Hankel matrix.

ROBUSTNESS STUDIES OF OUTPUT PREDICTIVE

DEAD-BEAT CONTROL FOR WING FLUTTER CONTROL APPLICATIONS

# I Introduction

## Background

The dead-beat control law for discrete systems is now well known and understood. Simply stated, the dead-beat control law assigns the discrete time closed loop eigenvalues to the origin. The states will be brought precisely to rest (assuming no additional disturbances) in no more than n (system order) discrete steps. Thus the dead-beat control law has been treated as an eigenvalue/eigenvector assignment problem (Ref 2, 4). The dead-beat control law anticipates the system response by feeding back all of the system states.

Another approach for anticipation of the systems response is to actually predict the system output into the future. Then, using this predicted output determine a control action which forces the predicted output identically to rest and remain at rest with no further required input. Output Predictive Dead-Beat Control (OPDEC) uses this approach in the formulation of its control law.

In the formulation of OPDEC, there is no restriction on the selection of the discrete sample rate. Theoretical analysis of OPDEC produced an approach for selection of an optimal sample rate, which might enhance the systems over-all robustness (Ref 4).

## Objectives

This report is concerned with the verification of the robustness properties, with respect to noise and model errors, of OPDEC, and not with the underlying theory used in the formulation of the control law.

The initial objectives of this report is to develop two programs for robustness verification. The first program finds the optimum sample rate of a given system. The second program implements OPDEC in a simulated closed loop environment with noise and model mismatch. After development of the programs and selection of a system to control, verification of OPDEC is then performed. Such items as analysis of robustness properties versus sample rate is a major objective. Checking the robustness properties of OPDEC when using a nonminimum phase system became another interesting area of concern. These objectives stated above are the prime areas of investigation of this report.

Potential Applications

This thesis is to provide a basis for possible future digital flight control applications of OPDEC. The OPDEC concept appears to be a good candidate for digital flight control applications because of its "robustness" properties. This characteristic is desirable because of dramatic changes in the B-52E's wing flutter dynamics with changing flight conditions (altitude and airspeed). The mathmatical models that describe the wing flutter dynamics of the B-52E (Ref 6) will be shown later.

## II Theory

An important class of control problems is the so called "tracking problem". The closed loop "tracking problem" roughly is the following. For a given reference variable, find an input such that the controlled system output follows or tracks the reference. A class of tracking problems consist of those where the reference variable is a constant. Such a problem is called the regulator problem (Ref 1: Ch 2).

With rapid development and miniaturization of digital computers, their use in control systems has become very common. A discrete controller that solves the regulator problem is the "dead-beat" controller. The "dead-beat" controller drives any initial state to zero in (at most) n steps, where n is the system order. The states, however, will not be driven to zero if the output is driven to zero (Ref 4: Ch 13). The output might have an unacceptable response between sample periods.

The Output Predictive "Dead-Beat" Controller (OPDEC) derived by Reid (Ref 3) has an acceptable response between sample rates when the output is driven to zero. The OPDEC control law will drive the output from any initial point to zero in at most n discrete steps. Because of the formulation, the output will also remain at zero unless disturbed. This means the "states" of the system are actually driven to zero in these n discrete steps. The needed sequence of control are obtained from (Ref 3)

$$U(k) = -(0.,0., \quad ,1)\cdot H_n^{-1}\cdot \underline{Y}(k+n/k-1) \qquad (1)$$

$$k = 0,1,2,\ldots,n-1$$

where the output prediction vectors

$$\underline{Y}(k+n/k-1) = \begin{bmatrix} y(k+n/k-1) \\ y(k+n+1/k-1) \\ \vdots \\ y(k+2n-1/k-1) \end{bmatrix} \tag{2}$$

elements is the output of the system at discrete times k+n, k+n+1, ...., k+2n-1 due to the system states at time k, or inputs up to time k-1, and $H_n$ is the discrete time Hankel matrix of size n. The Hankel matrix, shown below

$$H_n = \begin{bmatrix} h(1) & h(2)\ldots\ldots & h(n) \\ h(2) & h(3)\ldots\ldots & h(n+1) \\ h(3) & h(4)\ldots\ldots & h(n+2) \\ \vdots & \vdots & \vdots \\ h(n) & h(n+1)\ldots & h(2n-1) \end{bmatrix} \tag{3}$$

has only 2n-1 separate elements. This is helpful when trying to implement OPDEC on a small computer. The components of $H_n$ are the discrete impulse response of the system to be controlled.

According to Reid (Ref 3) given a SISO system

$$\dot{\underline{x}}(t) = A \cdot \underline{x}(t) + B \cdot u(t) \tag{4}$$

$$\underline{x}(0) = \underline{x}_0 \tag{5}$$

with sampled output (sample time T)

$$y(kT) = C \cdot \underline{x}(kT) \tag{6}$$

that is completely observable and completely controllable, with a discrete time model of the system (4)-(6) denoted as

$$x(k+1) = F \cdot \underline{x}(k) + G \cdot u(k) \tag{7}$$

$$\underline{x}(0) = \underline{X}_0 \tag{8}$$

$$y(k) = C \cdot \underline{x}(k) \tag{9}$$

where

$$F = e^{AT} \tag{10}$$

$$G = (\int_0^T e^{A\tau} \cdot d\tau) \cdot B \tag{11}$$

output can be driven to zero in n steps using OPDEC to control the system.

The elements of the output prediction vector, $\underline{Y}(k+n/k-1)$, can be found by

$$y(m/k-1) = C \cdot F^{m-k} \cdot \underline{x}(k) \tag{12}$$

or by the discrete conclution summation

$$y(m/k-1) = \sum_{i=o}^{\infty} h(m-k+i) \cdot u(k-i) \tag{13}$$

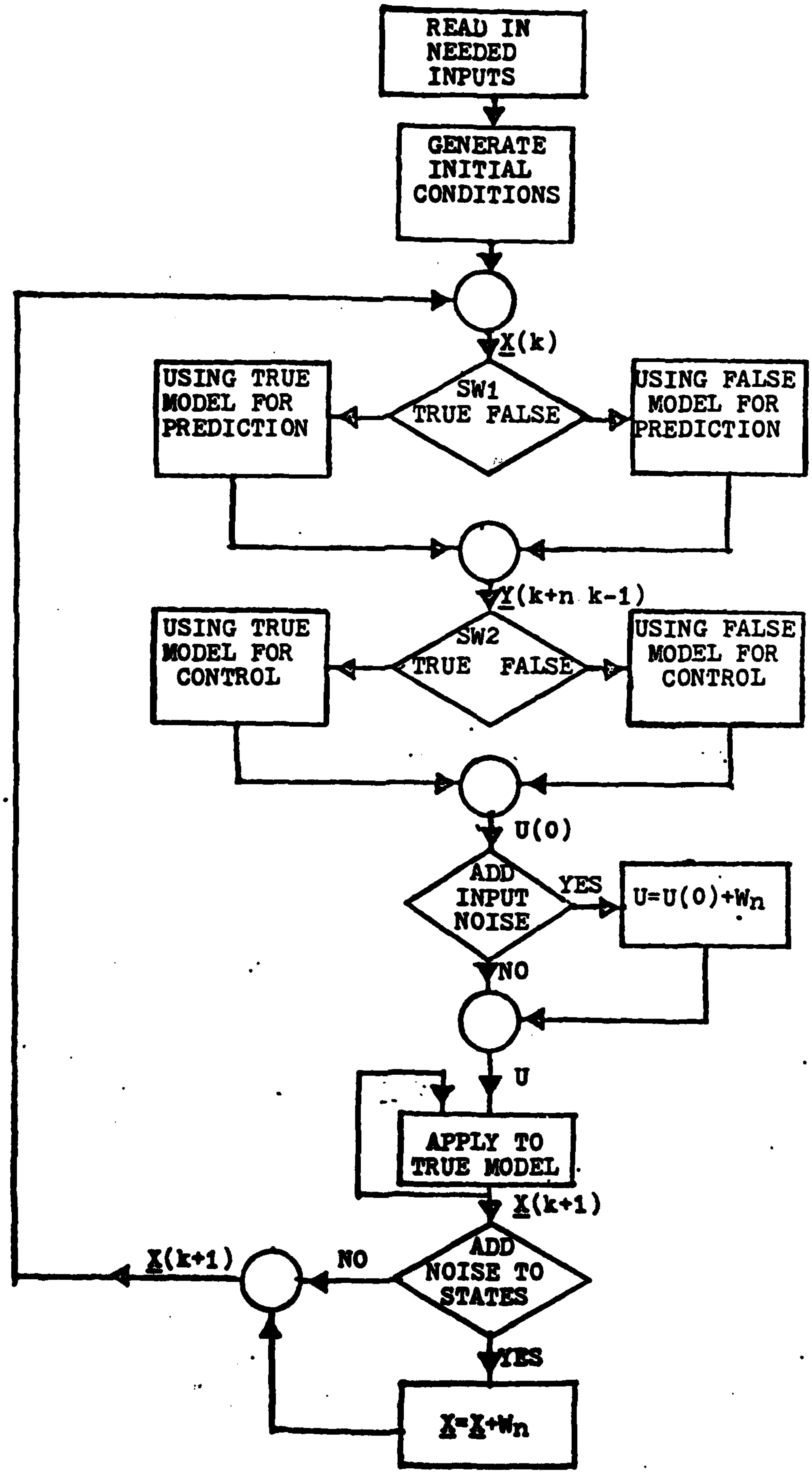
Since equation (12) can be used for prediction whether the open loop system is stable or unstable, this thesis will use equation (11) for output prediction. This will enhance future studies of OPDEC to controlling unstable systems.

Everything that has been discussed so far has not severely limited us in selection of the sample rate. The sample rate that minimizes the condition number of the Hankel matrix yields controls with good magnitude properties (Ref 4). This thesis is investigating if this same sample rate will yield good robustness characteristics as compared to other sample rates. This sample rate, in this report, is called the "optimal" sample rate.

## III Investigation

The investigation into the characteristics of OPDEC required two FORTRAN programs to implement the algorithm (See Appendices A and B). The first program implements a plot of the condition number versus the sample rate. This plot is used to find the optimal sample rate for the sampled data controller. This theoretical "optimum" occurs when the reciprocal condition number is a maximum (Ref 3). To verify the existance of an optimal sample rate with respect to robustness characteristics, the optimal sample rate and selected other sample rates were then used in the implementation of OPDEC.

The second program implements OPDEC and all the possible options needed to investigate the actual closed loop robustness properties of OPDEC (See Fig 1). This program is split into four general sections. The part initializes the program by reading in the inputs and generation of the initial state vector, $\underline{x}(o)$. Some of the inputs that are read in are the true and perturbed state matrix equations, sample rate and logic switches. The second part takes the state vector, $\underline{x}(k)$, and predicts the output in the future. These predictions are put in a vector format, y. This is done using either the true model as perturbed model according to the switch one (SW1) logic value. The third part takes the predicted output and uses the Hankel matrix, which it generates (Ref 3), to find the input, U. The Hankel matrix can be created using the true model or a perturbed model according to switch two's (SW2) logic value. The fourth part takes the input, U, and state vector $\underline{x}(k)$, and updates the state vector one sample time, $\underline{x}(k+1)$. Also, at this point, one can add noise to the input before it is applied to the true system. Also one can add noise directly to the updated states, x(k+1), before they are used in




**BLOCK DIAGRAM OF PROGRAM TWO** **Fig. 1**

the next prediction. The noise added in each case is a zero mean white guassian noise with selectable average strength. The programs are discussed in detail in appendix A and B.

## Models

This thesis used basically two math models. These models will be modified to help answer some basic robustness questions. The models are both reduced order models of the B-52E wing flutter modes (Ref 6). They both are very lightly damped. One is a fourth order system (Table 1) and the other is a tenth order system (Table 2). Most of the analysis was done on the fourth order system to save computer simulation time. The tenth order system was mainly used to see effects of reduced order models controlling larger systems.

TABLE 1

4th Order SISO System

Open-loop transfer function

$$\frac{8611.7698}{S^4+1.6S^3+274.075S^2+279.096S+8611.7698}$$

EIGENVALUES

-.55 + j 6.0
-.55 - j 6.0
-.25 + j 15.4
-.25 - j 15.4

## TABLE 2

### 10th Order SISO System

Open-loop transfer function

NUMERATOR

| POLYNOMIAL | ZEROS |
|---|---|
| ( -6607. )S**9 | ( .8112E-04) + j( .19443-01) |
| ( .2970E+06)S**8 | ( .8112E-04) + j( -.1944E-01) |
| ( -.2667E+07)S**7 | ( -1.843 ) + j( 4.255 ) |
| ( .1405E+09)S**6 | ( -1.843 ) + j( -4.255 ) |
| ( -.8689E+08)S**5 | ( 3.343 ) + j( -15.59 ) |
| ( .2036E+11)S**4 | ( 3.343 ) + j( 15.59 ) |
| ( .5901E+11)S**3 | ( -2.344 ) + j( 15.48 ) |
| ( .4132E+12)S**2 | ( -2.344 ) + j( -15.48 ) |
| ( -.4475E+08)S**1 | ( 46.64 ) + j( 0. ) |
| ( .1561E+09) | |

DENOMINATOR

| POLYNOMIAL | POLES |
|---|---|
| ( 1.000 )S**10 | ( -.9234E-03) + j( .6579E-01) |
| ( 6.626 )S**9 | ( -.9234E-03) + j( -.6579E-01) |
| ( 948.6 )S**8 | ( -.6772 ) + j( -1.053 ) |
| ( 4540. )S**7 | ( -.6772 ) + j( 1.053 ) |
| ( .2910E+06)S**6 | ( -.5446E-01) + j( 15.51 ) |
| ( .8712E+06)S**5 | ( -.5446E-01) + j( -15.51 ) |
| ( .2919E+08)S**4 | ( -1.917 ) + j( 16.84 ) |
| ( .3885E+08)S**3 | ( -1.917 ) + j( -16.84 ) |
| ( .4425E+08)S**2 | ( -.6631 ) + j( 20.15 ) |
| ( .2493E+06)S**1 | ( -.6631 ) + j( -20.15 ) |
| ( .1907E+06) | |

### Questions to be Addressed

The emphasis of this thesis is to investigate robustness properties of OPDEC with respect to model errors and noise addition. The obvious question to ask is how robust is OPDEC, and what factors enhance the robustness properties of OPDEC. The idea of an optimal sample rate naturally raises the question of the existance of an optimum. Since the Hankel matrix is instrumental in the selection of the optimum sample rate and in the usage of OPDEC, it would be interesting to see what sensitivities the Hankel matrix has. It would also be interesting to examine the issue of performance and robustness for non-minimum phase

systems. Controlling a system, with zeros in the right half plane, in the past has been difficult.

<u>Procedure and Results with 4th Order SISO System</u>

The following results use the same basic fourth order math model described previously. In some sections there has been modifications to this basic system. These modifications were needed to answer some specific questions and will be discussed in each section.

The system is assumed to start at a random initial state of

$$\underline{x}(o) = \begin{bmatrix} 7.049 \\ 8.095 \\ 5.007 \\ 9.745 \end{bmatrix}$$

in phase variable coordinates. For convenience, this coordinate system was selected and due to time considerations, this initial state was to only initial starting point used. The open loop response of the system from this initial state is shown in Figure 2.

<u>Selection and Implementation of Sample Rates</u>

The basic fourth order system was analyzed with the first program (Appendix A). From this analysis of the reciprocal condition number (See Fig 3) the optimum sample rate occurs at $\tau = .152$, with a maximum value of $1/K = .32356$.

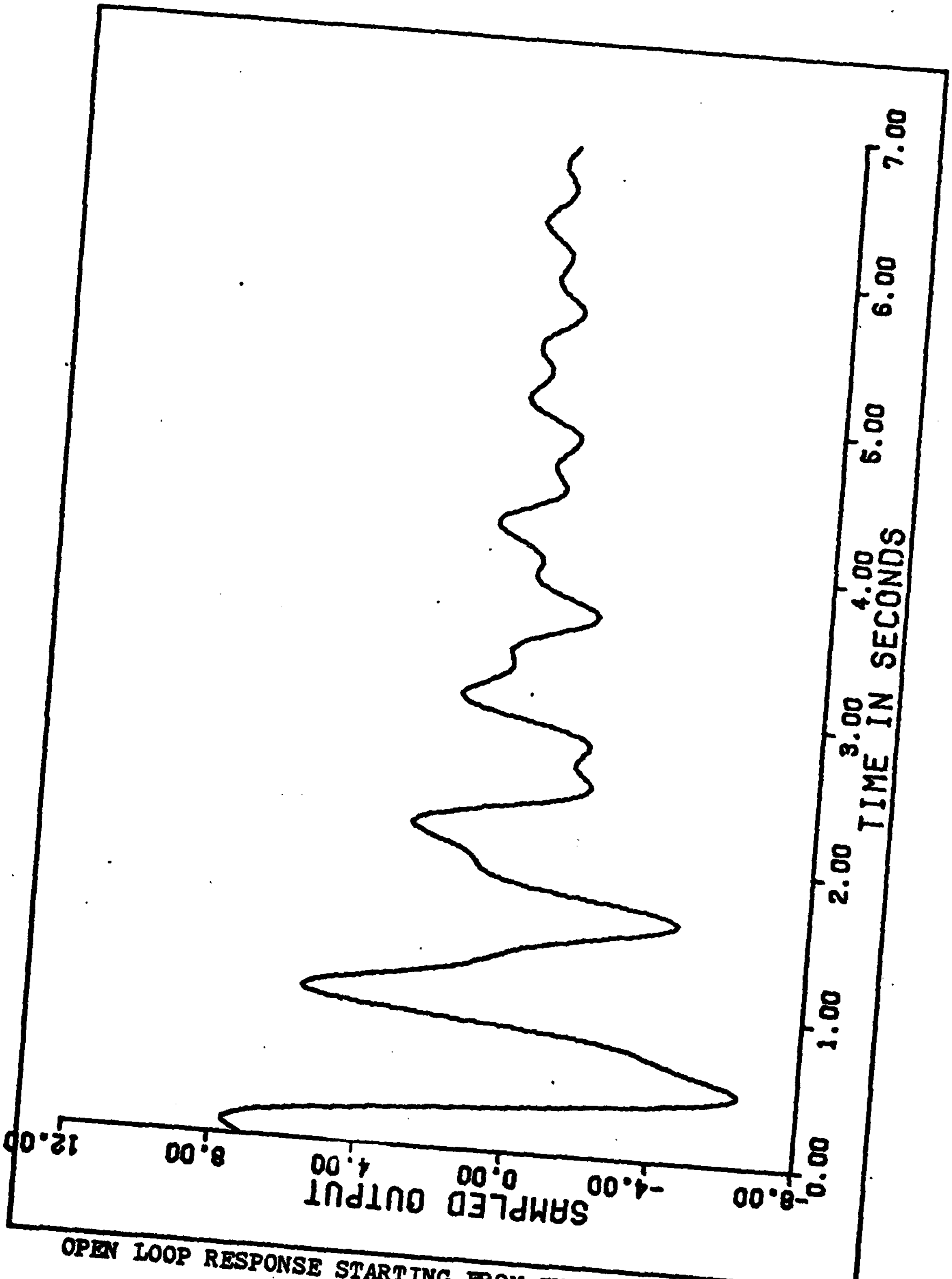
Besides this optimum sample rate, it was desired to compare performance at several other selected sample rates. An interesting sample rate to look at is $\tau = .231$. This sample rate occurs at the relative maximum of the second lobe. This sample rate was chosen to see if relative optimization occurs. Two other sample rates chosen are .113 and .182. They were chosen because their condition numbers were the same as the sample rate of .231. This was done to see if the value of

the condition, no matter where it occurs, can be used to predict the robustness of the system. Another sample rate chosen was .954 because it occurs at the relative maximum of the last lobe. It was chosen because its condition number is close to the value of the previous three. The sample rates chosen, their reciprocal condition number and their condition numbers are listed in Table 3. The next three sample rates were chosen because of their small condition numbers. The sample rates are .085, .214 and .5. The last three sample rates were chosen because of their very small reciprocal condition numbers. The sample rates are .04, .623 and .835. These choices of sample rates appear to cover the broad range of the condition number and should provide enough information for this study.

TABLE 3

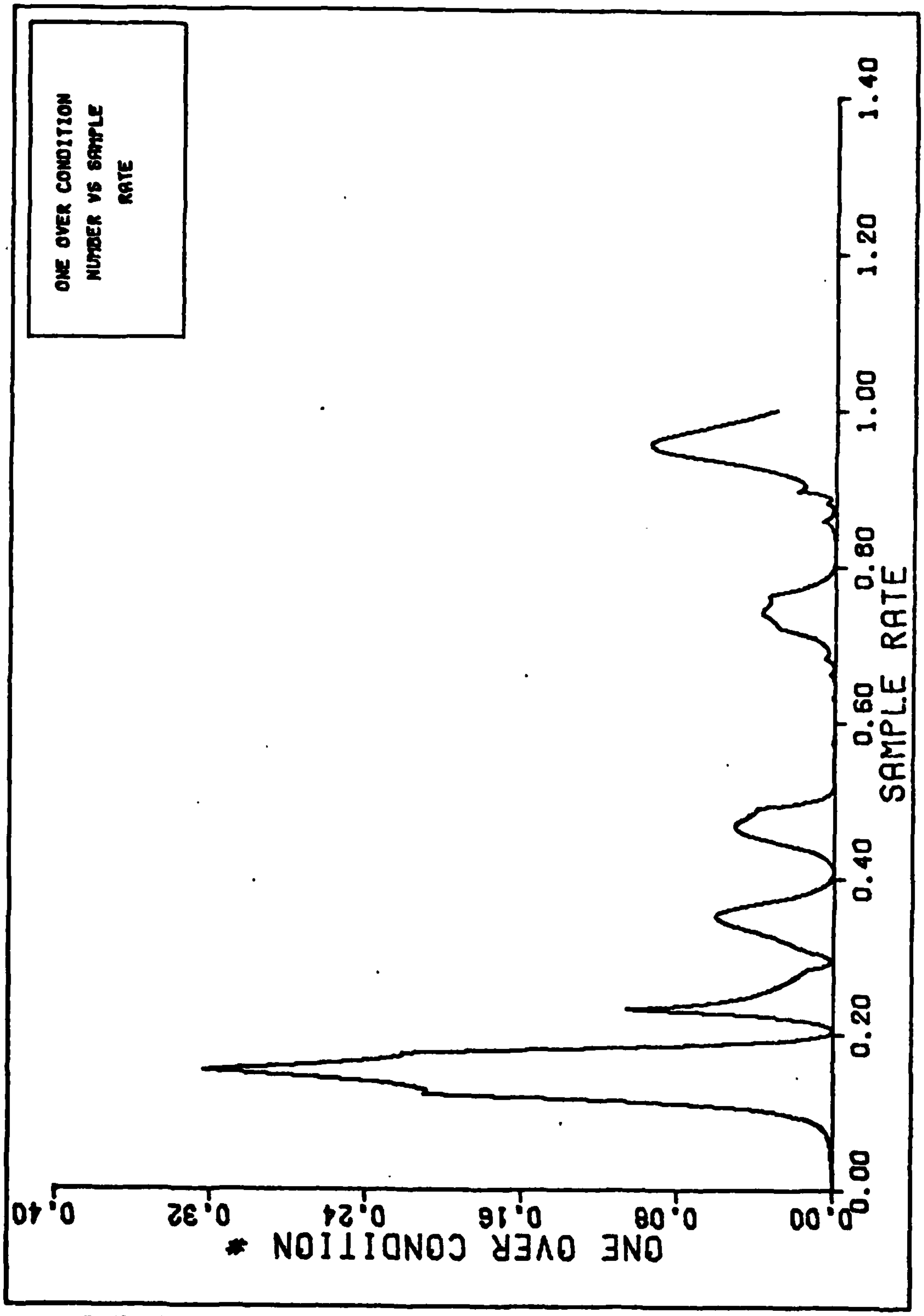
| SAMPLE RATES CHOSEN | 1/K | K |
|---|---|---|
| .152 | .32356 | 3.090 |
| .231 | .10646 | 9.392 |
| .182 | .10714 | 9.333 |
| .113 | .10664 | 9.377 |
| .954 | .09551 | 10.469 |
| .085 | .00925 | 108.101 |
| .214 | .00930 | 107.511 |
| .5 | .00931 | 107.366 |
| .04 | .0004160 | 2403.557 |
| .623 | .00003552 | 28151.568 |
| .835 | .000423 | 2363.574 |

The plots of the output predictive response and the sequence of control inputs for selected sample rates are shown in Figures 4 thru 16. These plots were generated using the true model in both prediction and control phases and no noise added. Figures 4 and 5 are the output




OPEN LOOP RESPONSE STARTING FROM INITIAL STATE, $\underline{X}(0)$

**Fig. 2**

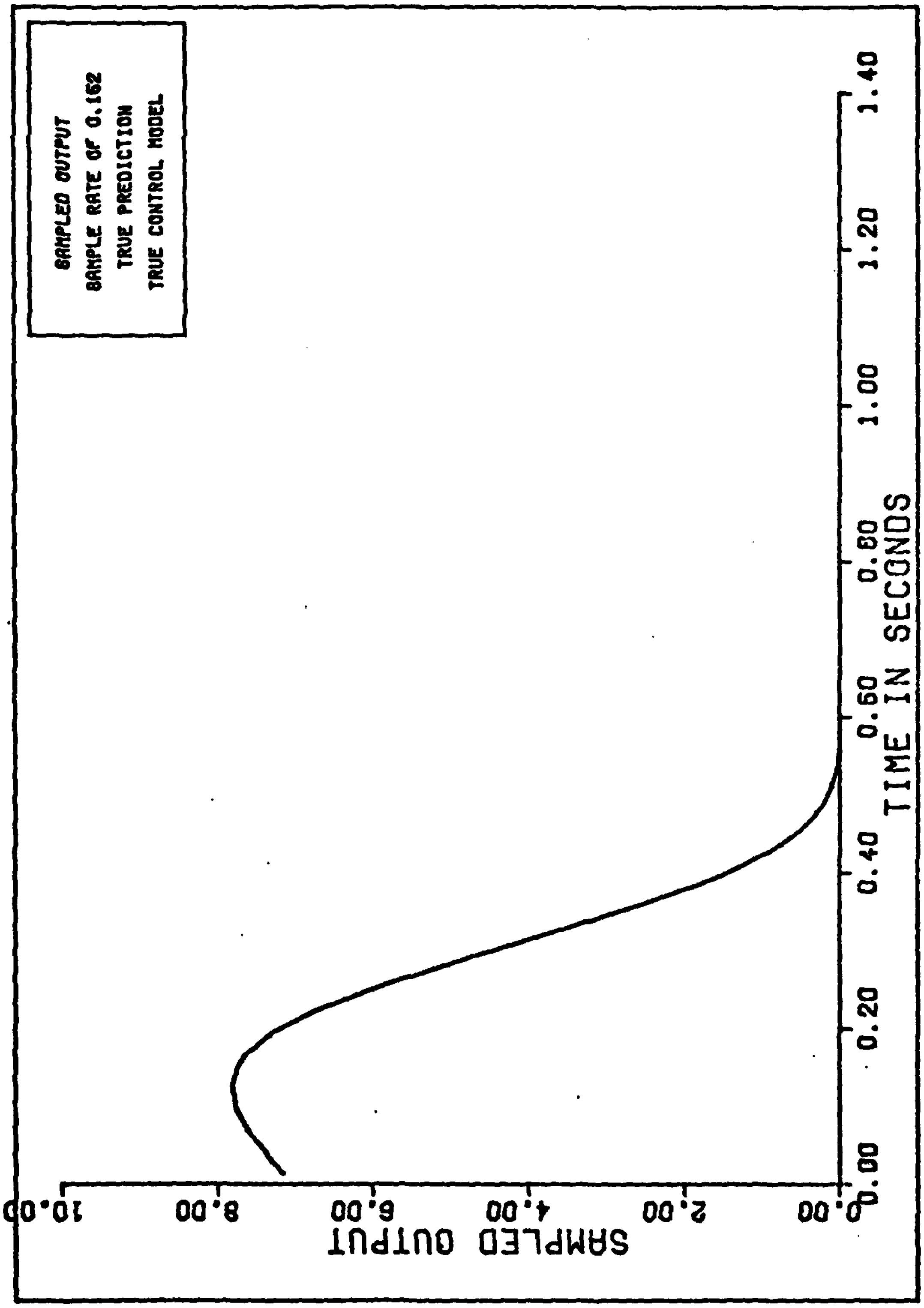



ONE OVER CONDITION NUMBER VS SAMPLE RATE

Fig. 3

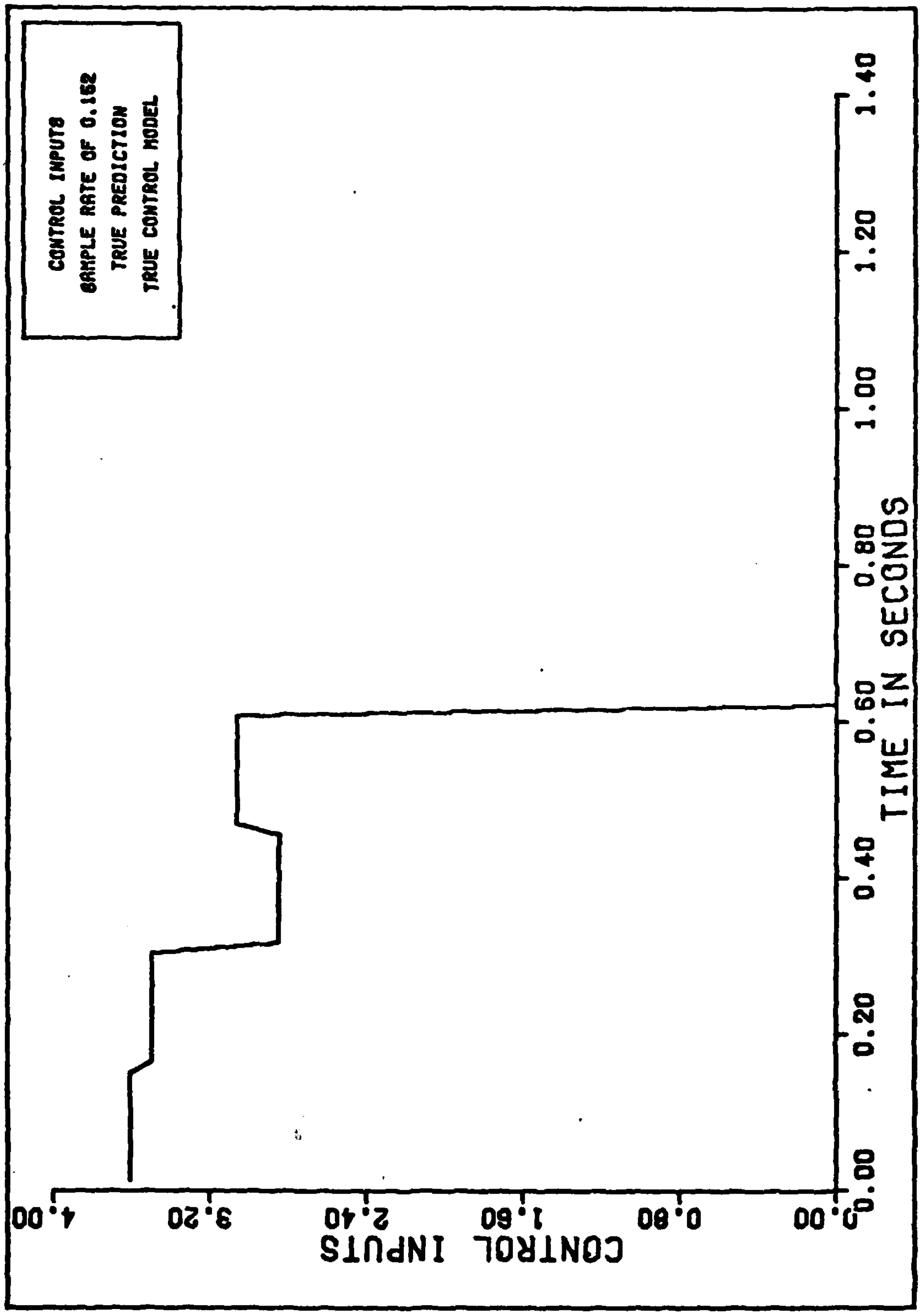
response and control inputs at the optimum sample rate (.152). The output response and control inputs for sample rates of .113, .182 and .231 are shown in Figures 6 and 7. The output response for sample rates of .085, .214 and .5 are shown in Figure 8. The control inputs for sample rates of .5 and .214 are shown in Figure 9. Due to the large scale needed, the control inputs for the sample rate of .085 is shown separate in Figure 10. The output response and control inputs for the sample rate of .04 are shown in Figures 11 and 12. Looking at Figures 10 and 12, the control inputs for two "poorly conditioned" sample rates, one notices the inputs getting larger as the reciprocal condition number is getting smaller. This is what was expected from the theory (Ref 3). Figures 13 and 14 are the output response and control inputs for the sample rates of .623 and .954. The output response and control inputs for the sample rate of .835 is shown in Figures 15 and 16.

Studying the Figures, one notices some trends. First, as the sample time increases the output response starts fluctuating more. This is because the sample rate is slower than the natural frequency of the system and the system has more time to fluctuate. As the sample rate gets small, the value of the control inputs increase. This is shown by Figure 7. Thirdly, the condition number has some effect on the size of the control inputs. When both small reciprocal condition numbers and small sample rates combine, the control input becomes huge (Fig 12). Notice the magnitude difference in control input required between sample rate of .152 and .04. This is caused by the small time increment the system has to achieve the dead-beat response. For the 4th order system, the system comes to rest in 4 steps regardless of the sample rate. Thus with a high sample rate the system has to work much harder to drive the states to zero in the four steps. This is the major cause of magnitude difference.



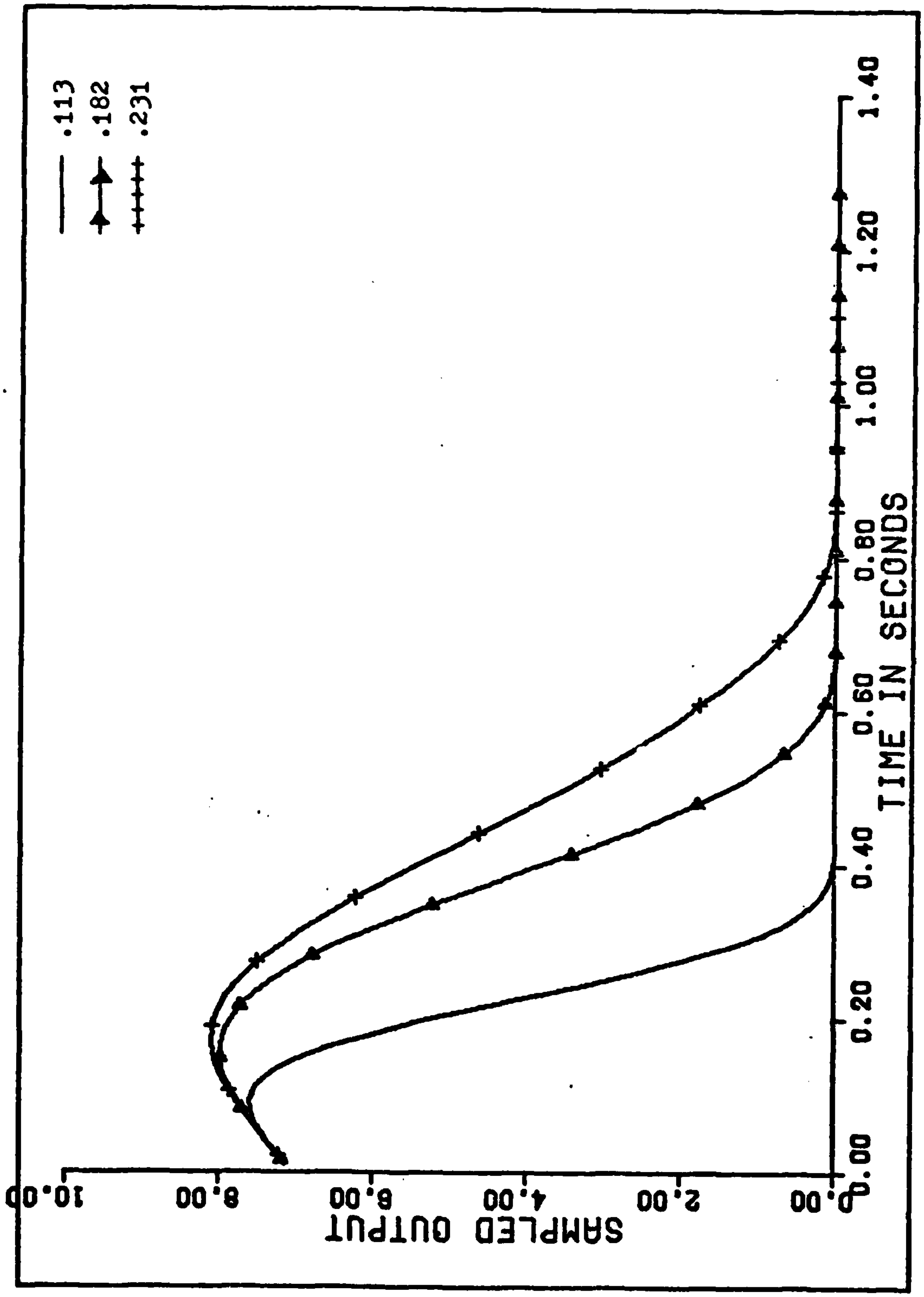

SAMPLED OUTPUT WITH NO NOISE ADDED
AT A SAMPLE RATE OF .152 SECONDS

Fig. 4




CONTROL INPUTS WITH NO NOISE ADDED
AT A SAMPLE RATE OF .152 SECONDS

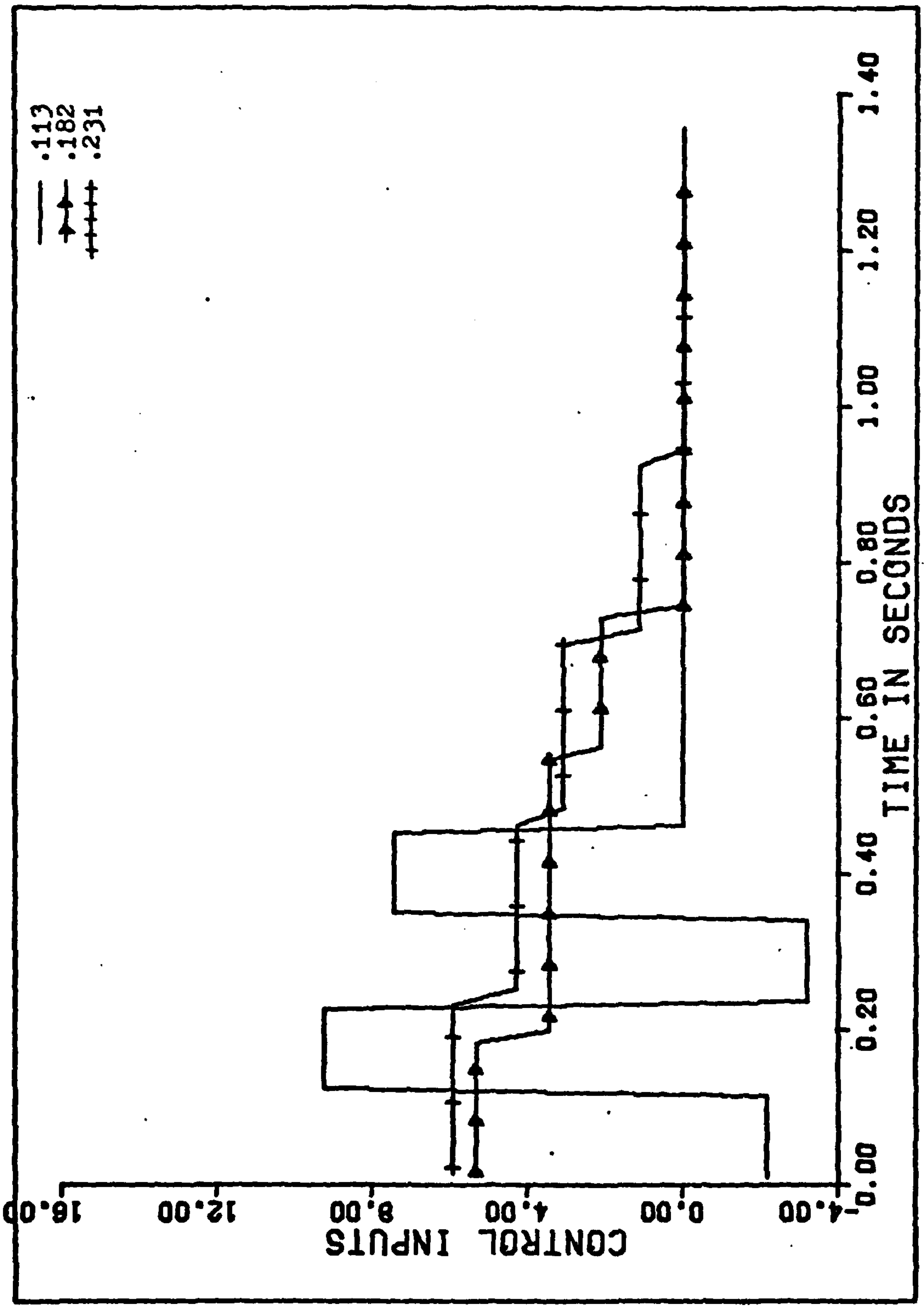
Fig. 5




SAMPLED OUTPUT WITH NO NOISE ADDED AT
SAMPLE RATES OF .113, .182, and .231 SECONDS



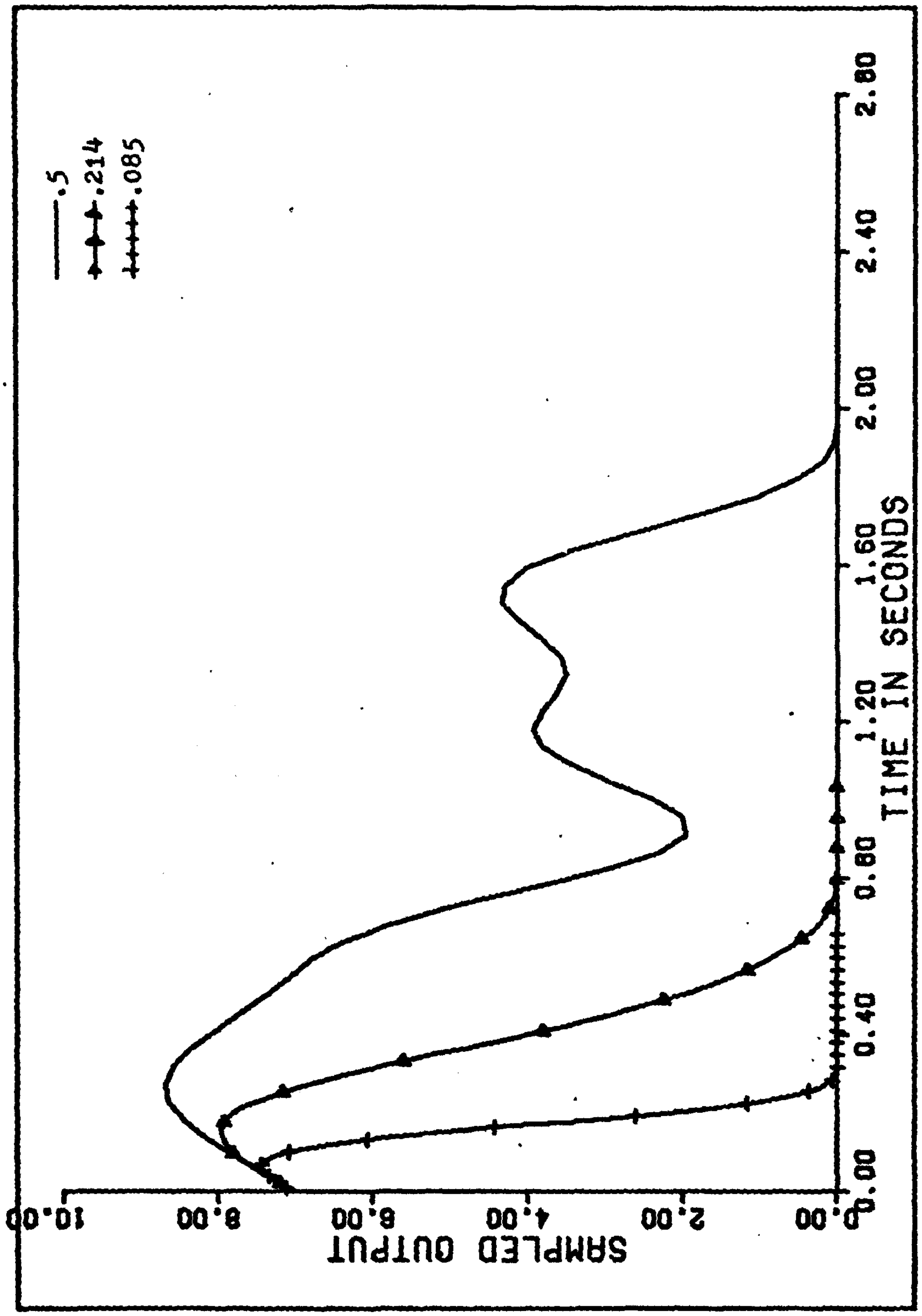
Fig. 6




**CONTROL INPUTS WITH NO NOISE ADDED AT SAMPLE RATES OF .113, .182, and .231 SECONDS**

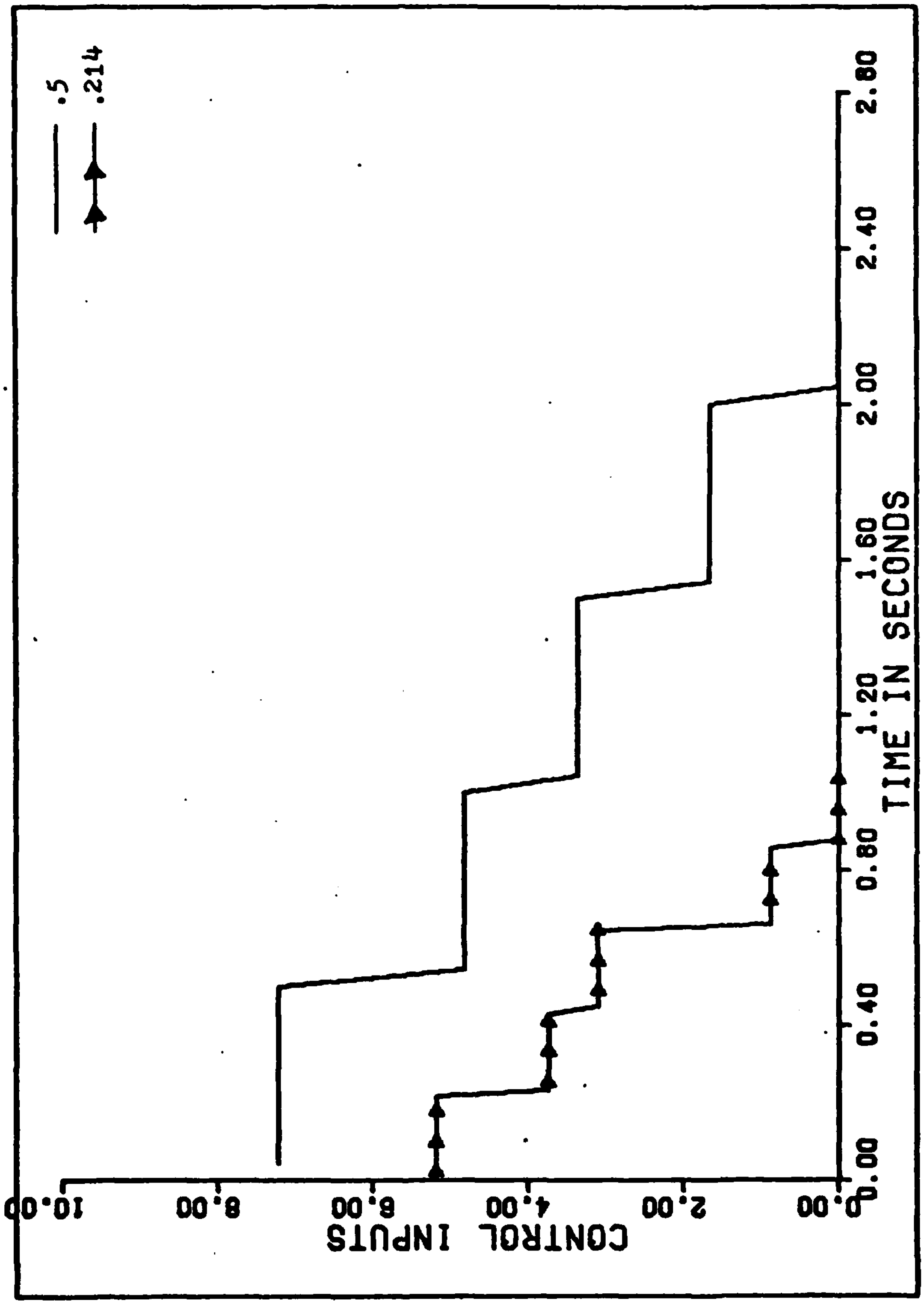


**Fig. 7**




SAMPLED OUTPUT WITH NO NOISE ADDED AT
SAMPLE RATES OF .5, .214 and .085 SECONDS

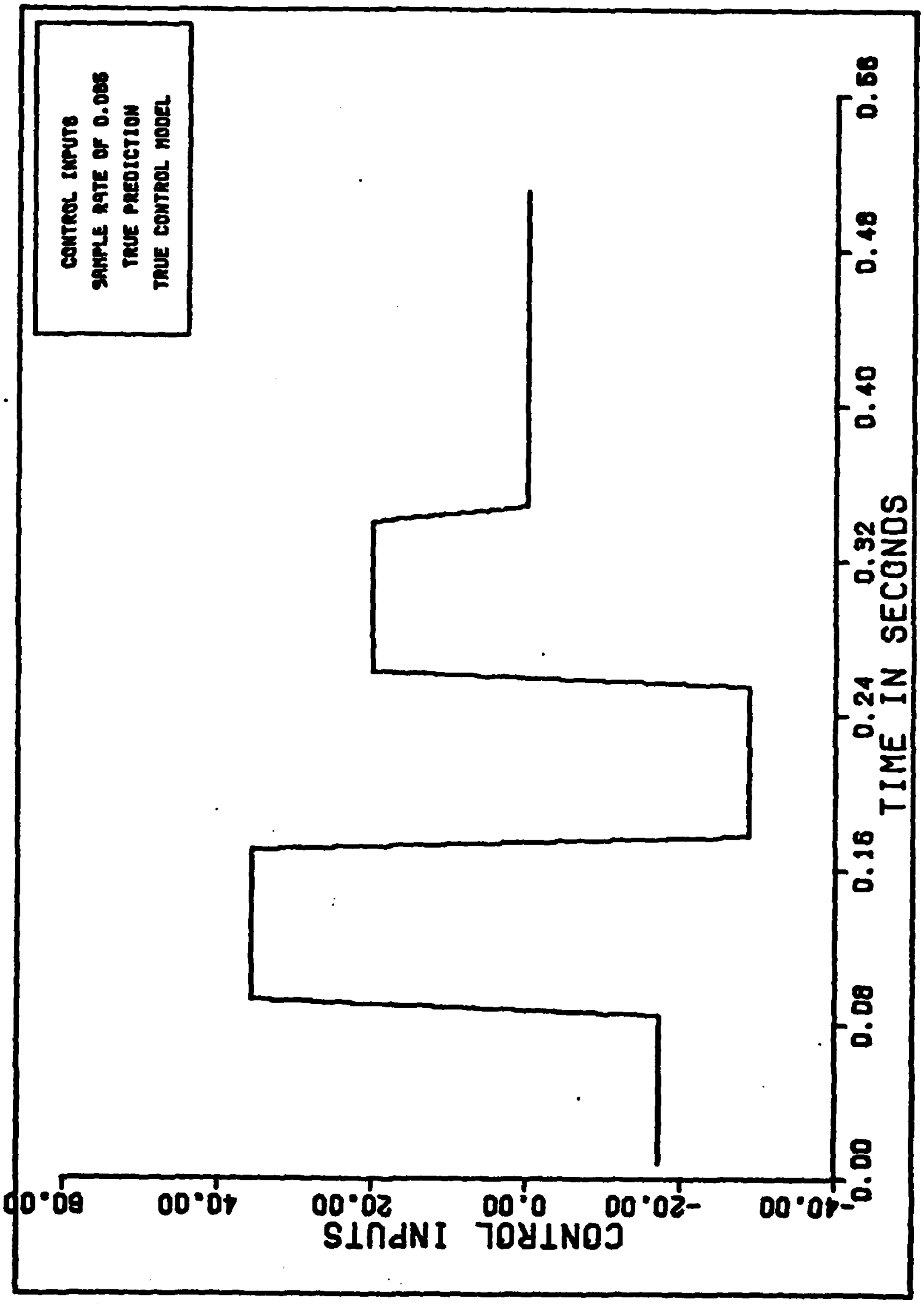
Fig. 8




CONTROL INPUTS WITH NO NOISE ADDED AT
SAMPLE RATES OF .5 and .214 SECONDS



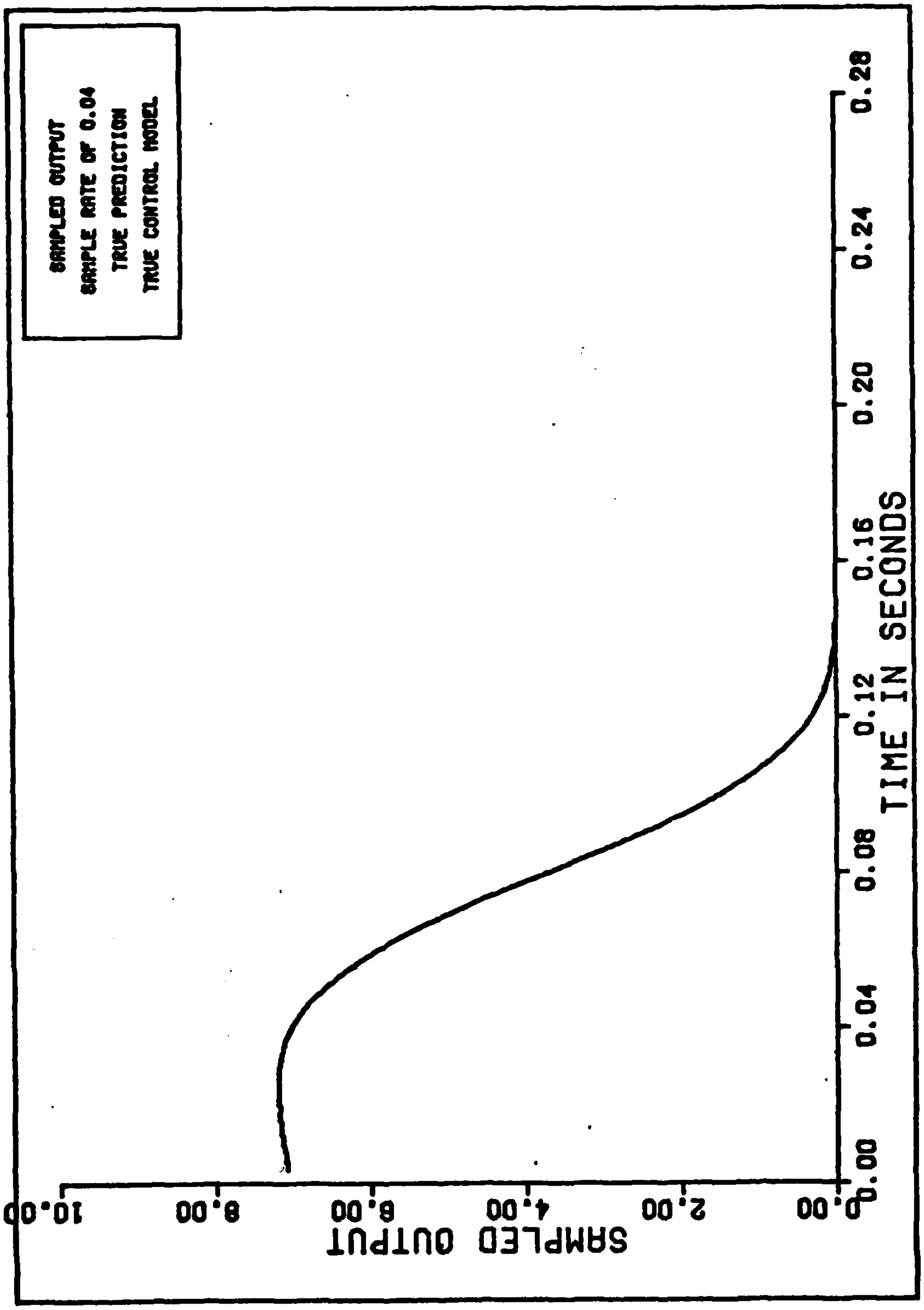
Fig. 9




CONTROL INPUTS WITH NO NOISE ADDED AT A SAMPLE RATE OF .085 SECONDS

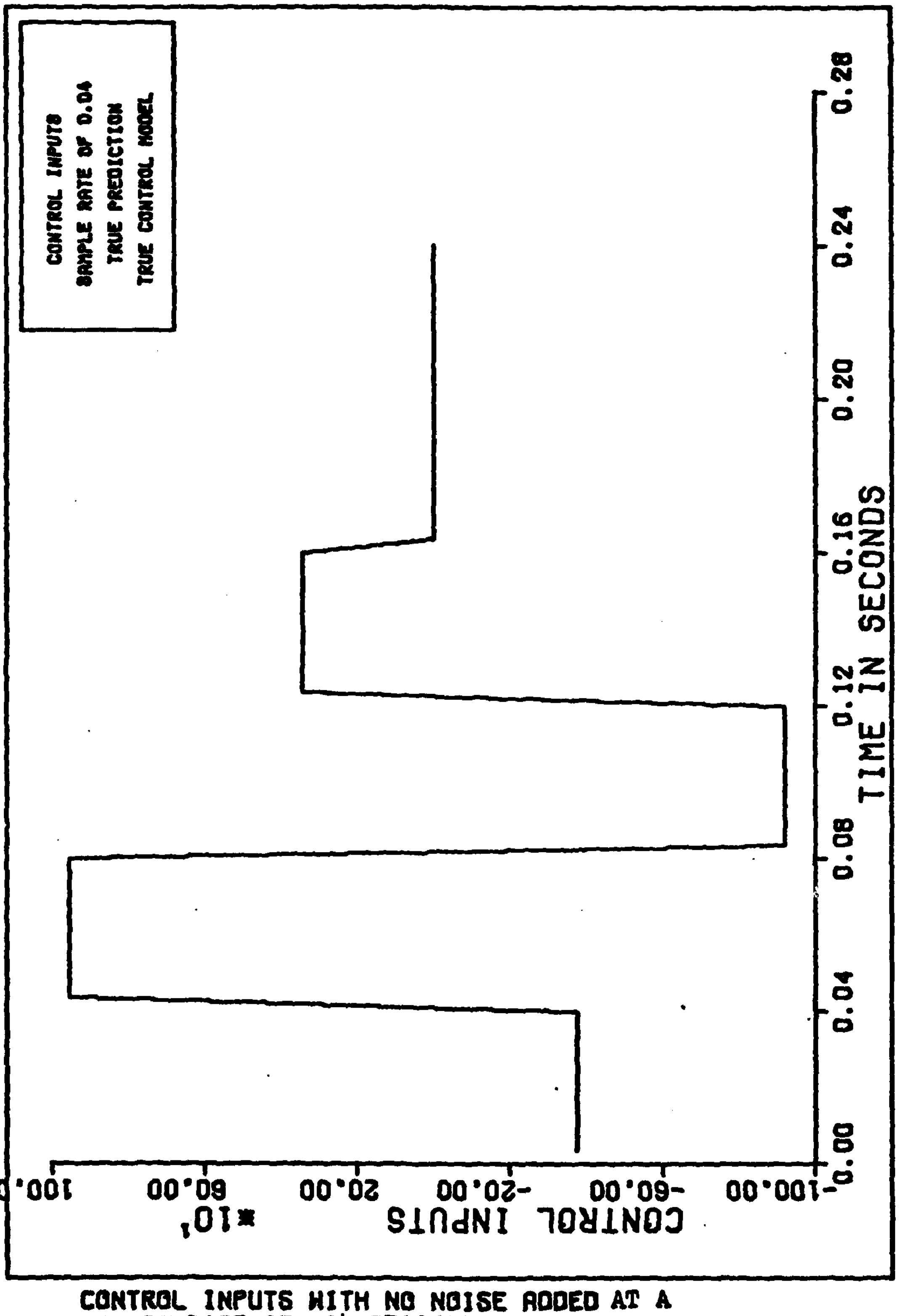


Fig. 10



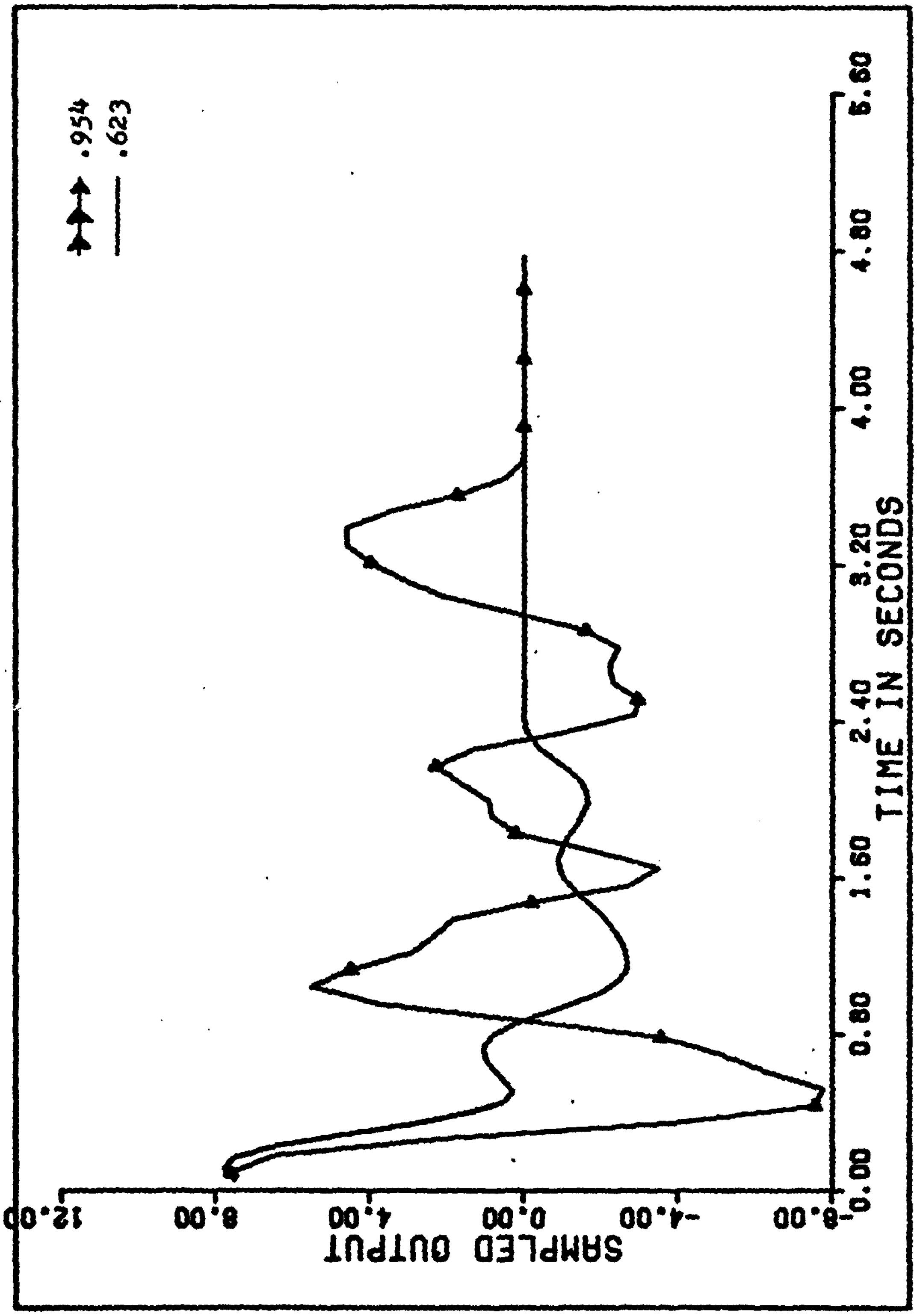

SAMPLED OUTPUT WITH NO NOISE ADDED AT A
SAMPLE RATE OF .04 SECONDS

Fig. 11




CONTROL INPUTS WITH NO NOISE ADDED AT A SAMPLE RATE OF .04 SECONDS

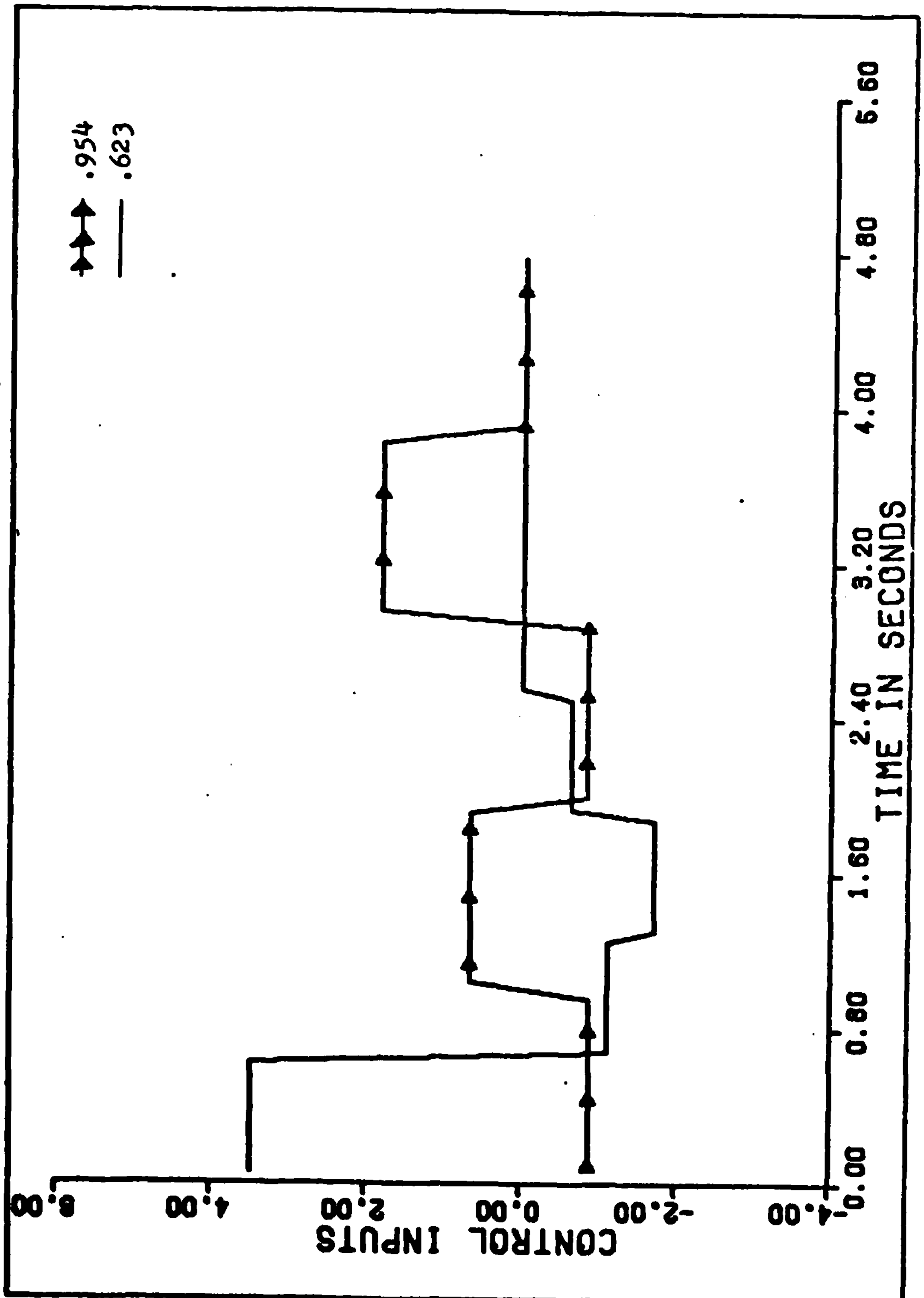
Fig. 12




SAMPLED OUTPUT WITH NO NOISE ADDED AT
SAMPLE RATES OF .954 and .623 SECONDS



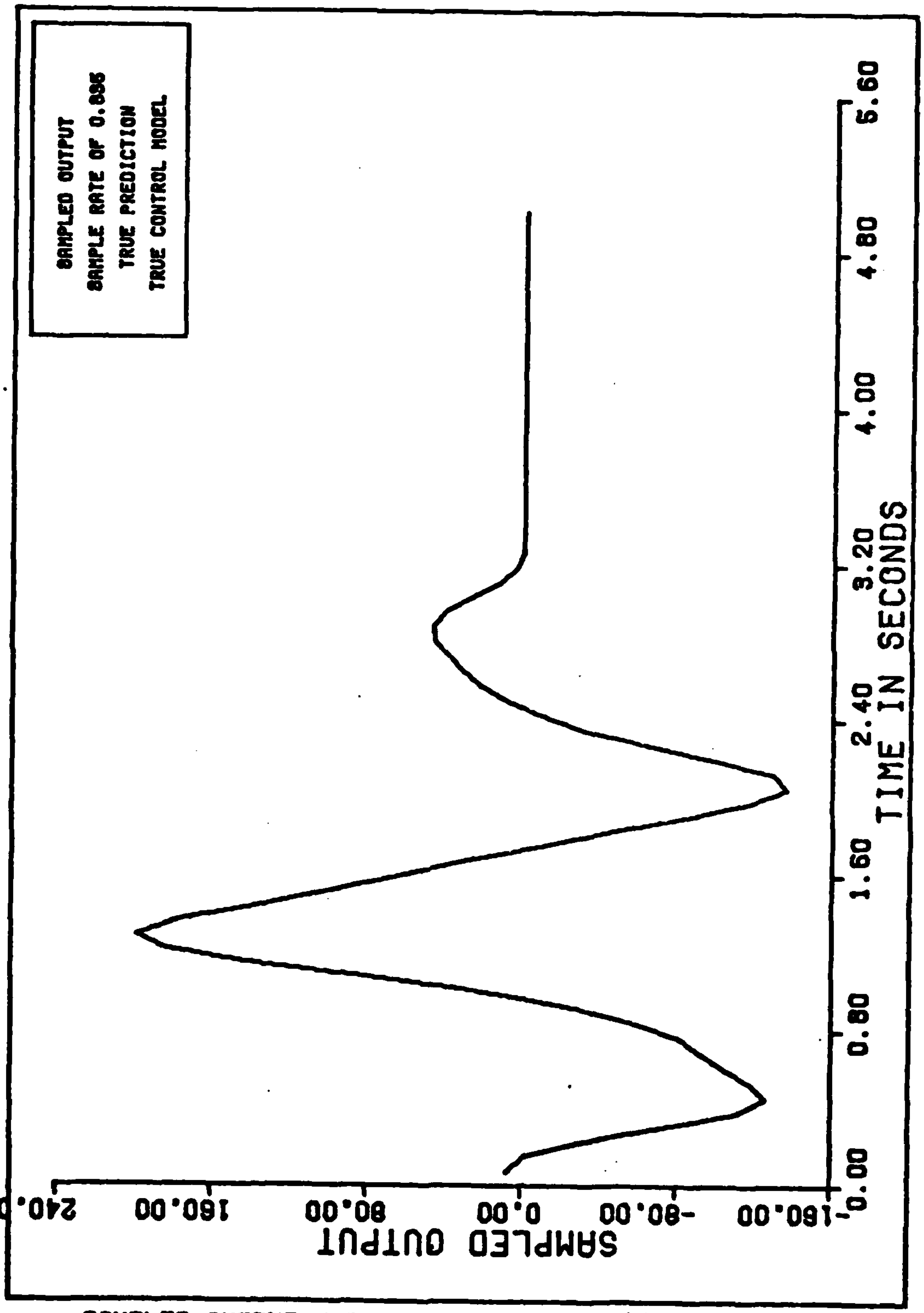
Fig. 13




CONTROL INPUTS WITH NO NOISE ADDED AT
SAMPLE RATES OF .954 and .623 SECONDS



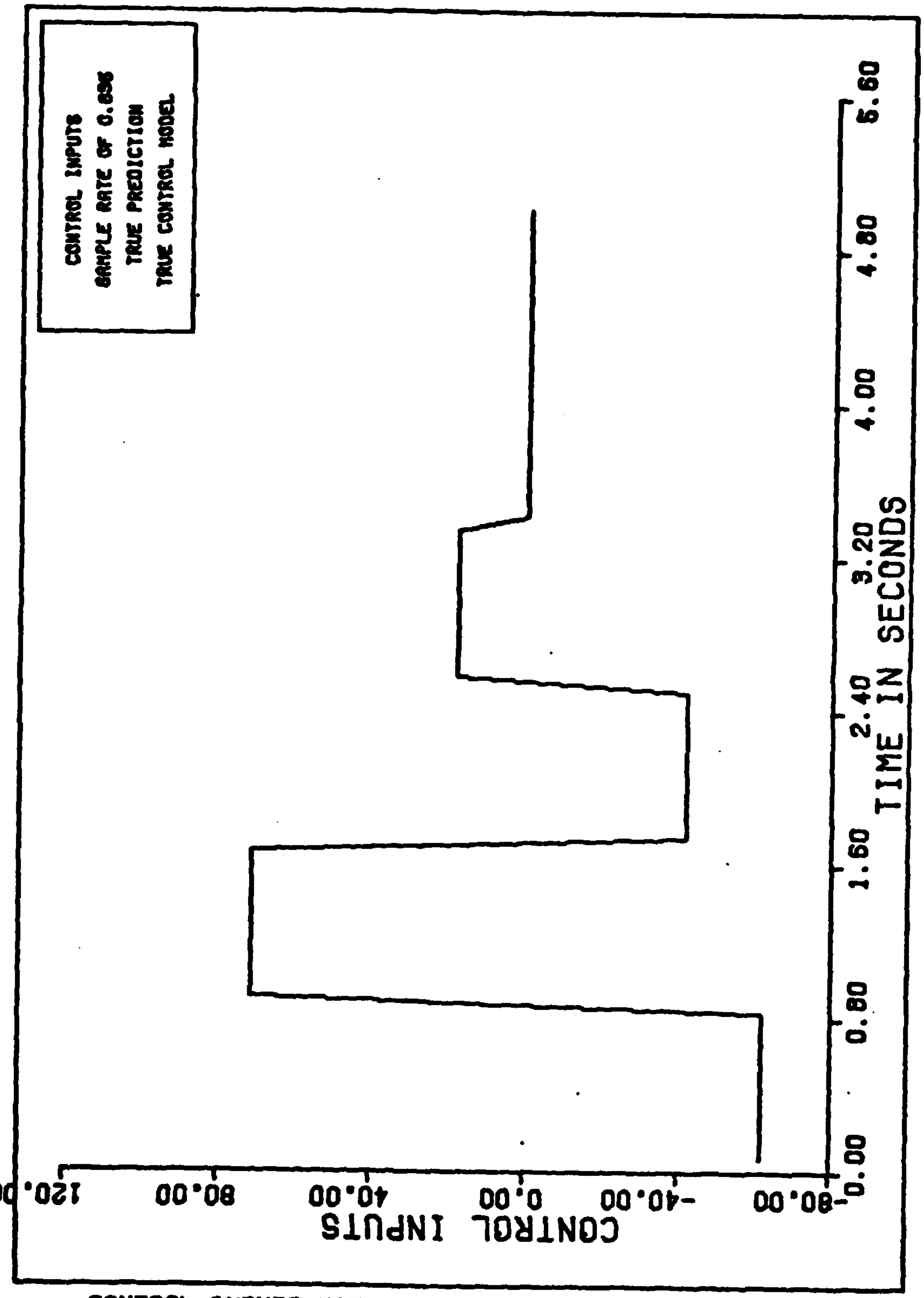
Fig. 14




SAMPLED OUTPUT WITH NO NOISE ADDED AT A
SAMPLE RATE OF .835 SECONDS



Fig. 15




CONTROL INPUTS WITH NO NOISE ADDED AT A SAMPLE RATE OF .835 SECONDS.



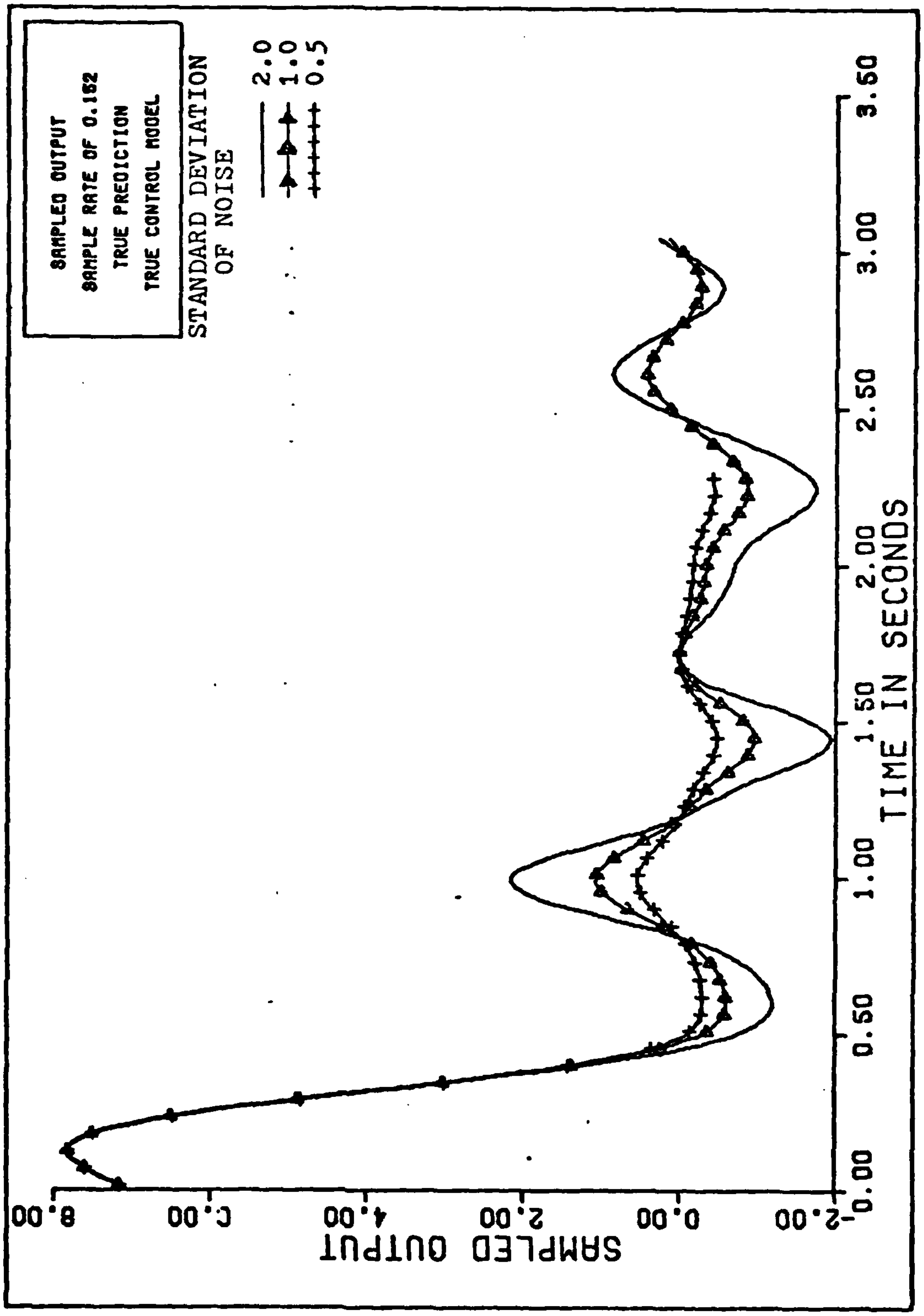
Fig. 16

## Effect of Noise

This section implements OPDEC with input and/or state noise added. This is done for all the sample rates selected. The average strengths of the noises added in either case (input noise or state noise) are .5, 1.0, 1.5 and 2.0. The sampled output responses, for all selected sample rates, fell into 3 categories. They are stable, conditionally stable and unstable. Conditionally stable means the output is varying around the final value and the output is never larger than it would be when no noise is added. Each section has a table that consolidates the output result.

## Adding Input Noise

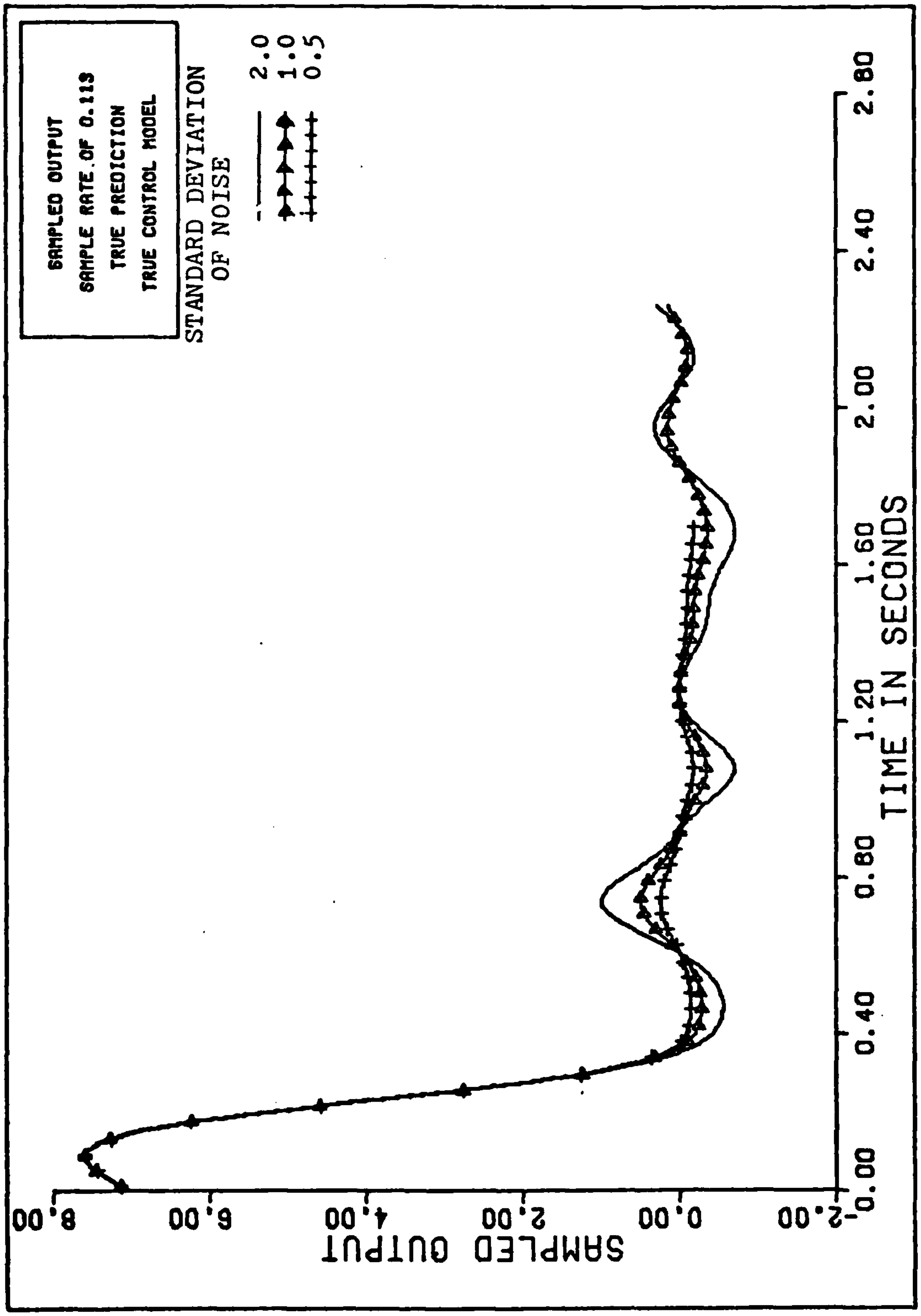
In this section, noise is added eleven times to the input while it is being input into the true system. What one would expect is that as the average value of the input noise is increased, the output response would become worse. This can be seen in Fig 17 and 18 which have three plots per figure at the same sample rate with different average input noises. Table 4 shows the effects of input noise on the sampled output at all the selected sample rates.




SAMPLED OUTPUT WITH INPUT NOISE ADDED AT A
SAMPLE RATE OF .152 SECONDS



Fig. 17




SAMPLED OUTPUT WITH INPUT NOISE ADDED AT A SAMPLE RATE OF .113 SECONDS



Fig. 18

TABLE 4

Output Responses With Input Noise Added

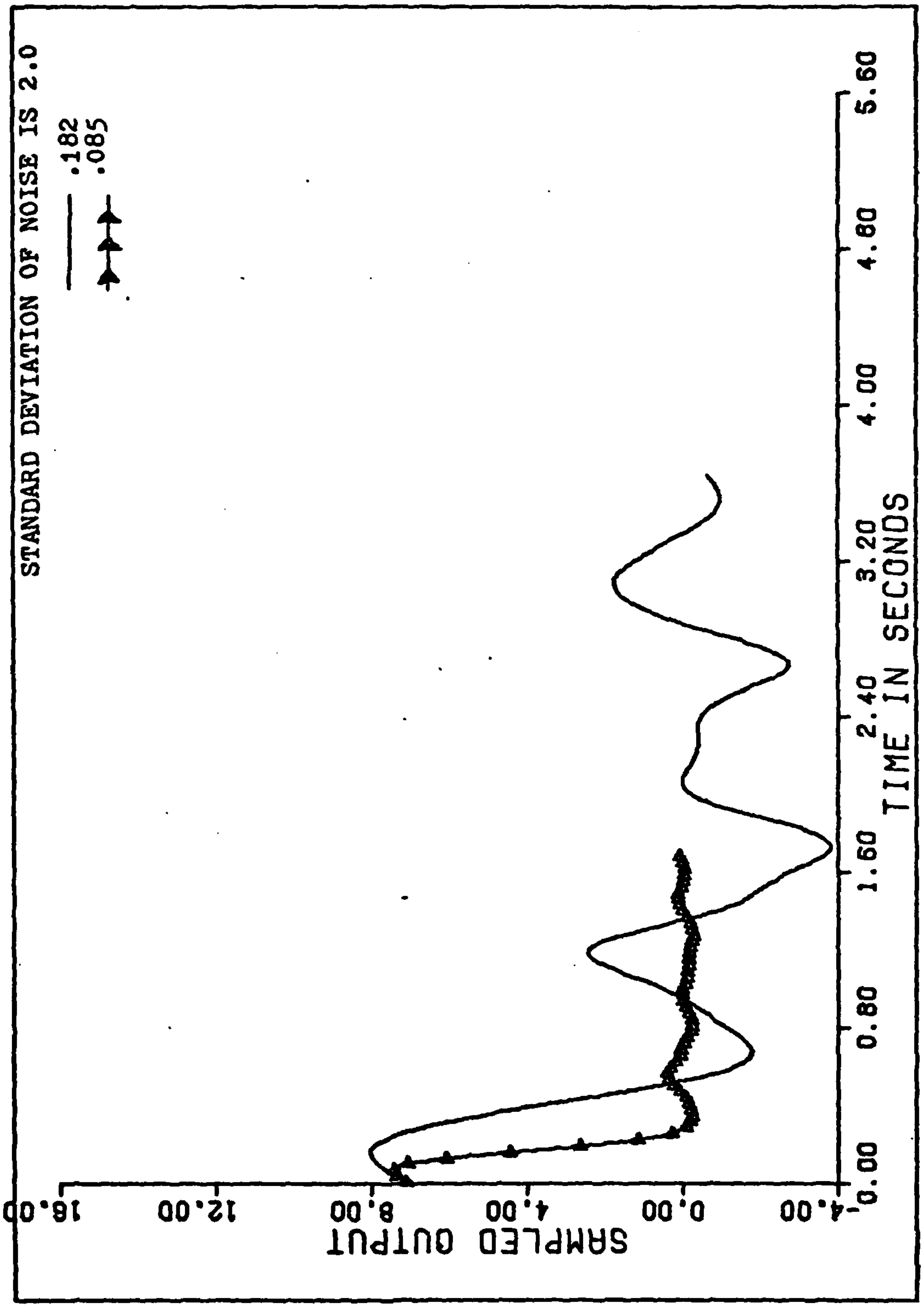
| SAMPLE TIME OF CONTROL | SAMPLE TIME OF INPUT NOISE | OUTPUT RESPONSE WITH INPUT NOISE STRENGTH OF | | | | 1/K |
|---|---|---|---|---|---|---|
| | | .5 | 1.0 | 1.5 | 2.0 | |
| .4 | .0036 | S | S | S | S | .000416 |
| .085 | .0077 | S | S | S | S | .00925 |
| .113 | .0102 | S | S | CS | CS | .10664 |
| .152 | .0138 | CS | CS | CS | CS | .32356 |
| .182 | .0165 | CS | CS | CS | CS | .10714 |
| .214 | .0194 | CS | CS | CS | CS | .0093 |
| .231 | .021 | CS | CS | CS | CS | .10646 |
| .5 | .0454 | CS | CS | US | US | .00931 |
| .623 | .0566 | US | US | US | US | .0000355 |
| .835 | .0759 | CS | CS | US | US | .000423 |
| .954 | .0867 | CS | US | US | US | .09551 |

S - STABLE

CS - CONDITIONALLY STABLE

US - UNSTABLE

From Table 4 and the control input plots in the previous section, one notices a trend. It seems the effects of input noise depends upon the magnitude of the control inputs. This can be seen in Fig 19 which shows the sampled output with input noise added at sample rates of .085 and .182. The output response at the "poorer conditioned" sample rate (.085) is better than the response at the sample rate of .182. This is because of the magnitude difference between the respective control inputs. It seems the larger the control the less sensitive the output response is to input noise. It would then be expected that the output response at the sample rate of .835, would be insensitive to input noise because of its large inputs relative to the optimal sample rate (Fig 16). Fig 20 is the output response with input noise added at a sample rate




SAMPLED OUTPUT WITH INPUT NOISE ADDED AT
SAMPLE RATES OF .182 and .085 SECONDS

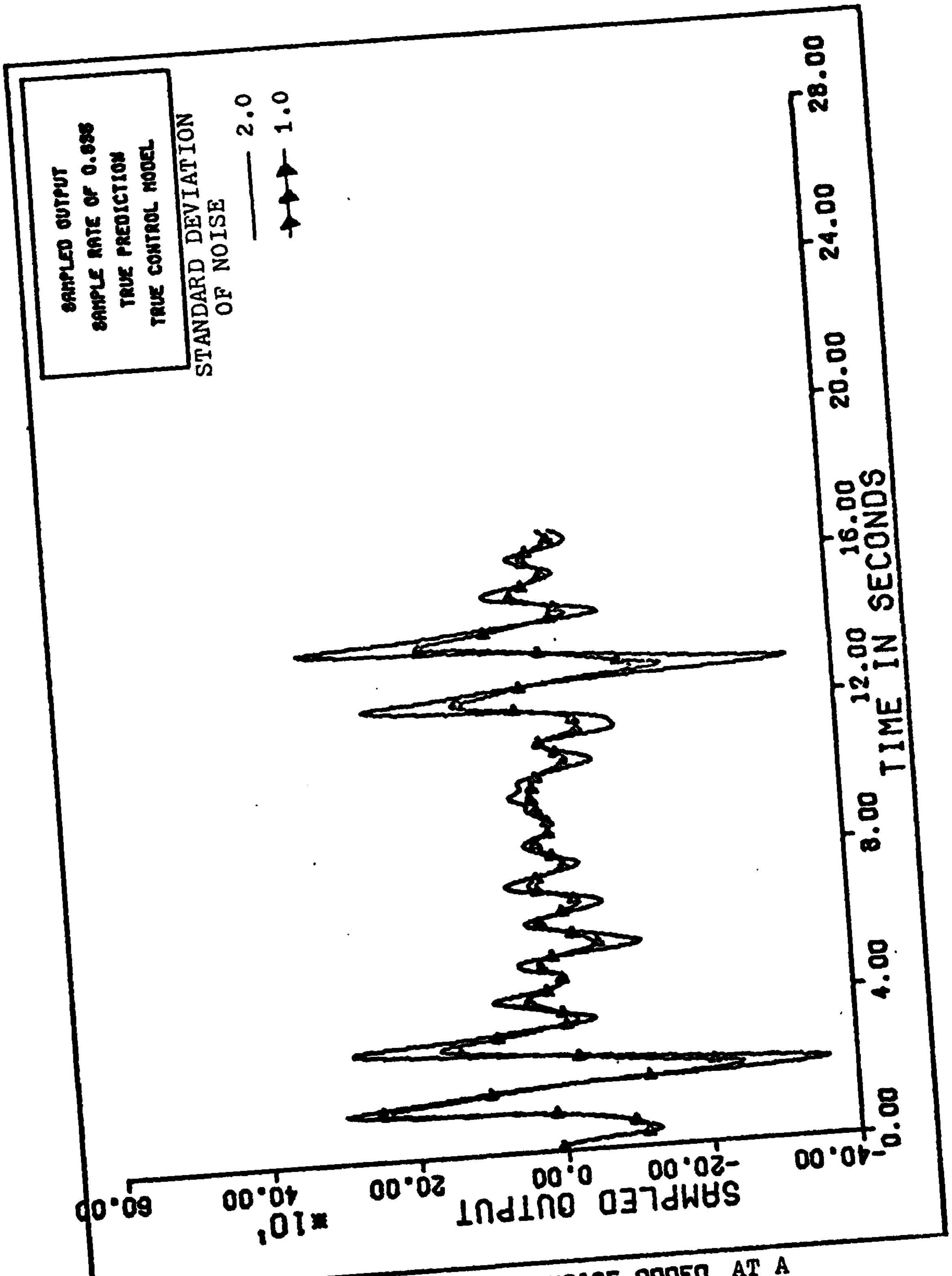


Fig. 19

of .835. This shows that the effects of input noise on the output is not solely determined by the relative magnitude of the control inputs or the condition number before the addition of the noise. It shows that the sample rate also effects the robustness of a system that has input noise added. Fig 21 is an example of the control inputs with input noise added, at a sample rate of .113 seconds.

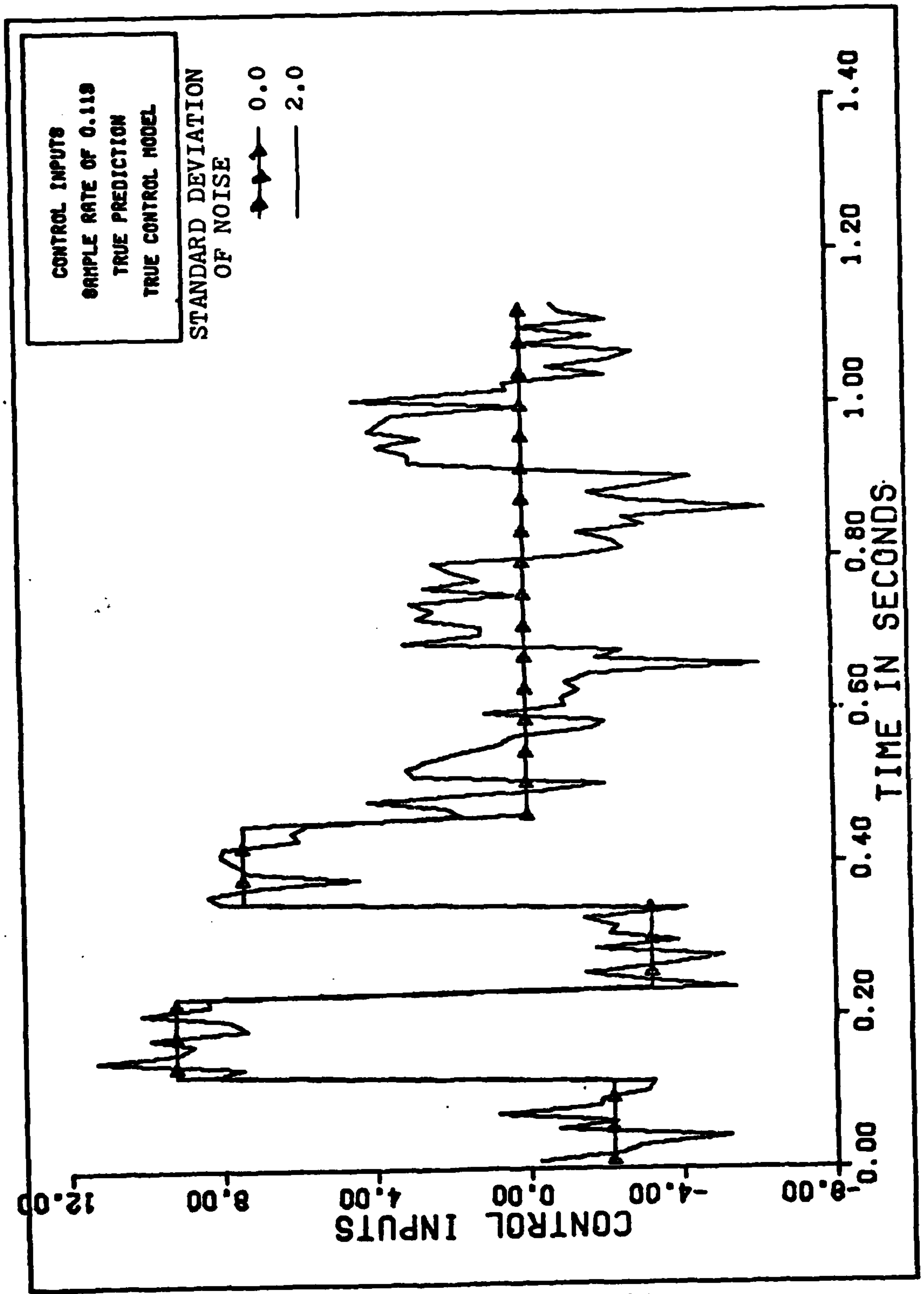
Adding State Noise

In this section, noise was added to all the states after they had been updated one sample step. This is a simplified simulation of the influence of incorrect state estimation from a Kalman filter in the closed loop controller. The addition of noise occurs just before the next output prediction. What one would expect is when the average strength of the noise is increased, the output response becomes worse. Fig 22, 23 and 24 each have three plots per figure at the same sample rate with different average noises corrupting the states. These figures verify our expectations. Table 5 shows the effects of state noise on the sampled output at all the selected sample rates.




SAMPLED OUTPUT WITH INPUT NOISE ADDED AT A
SAMPLE RATE OF .835 SECONDS

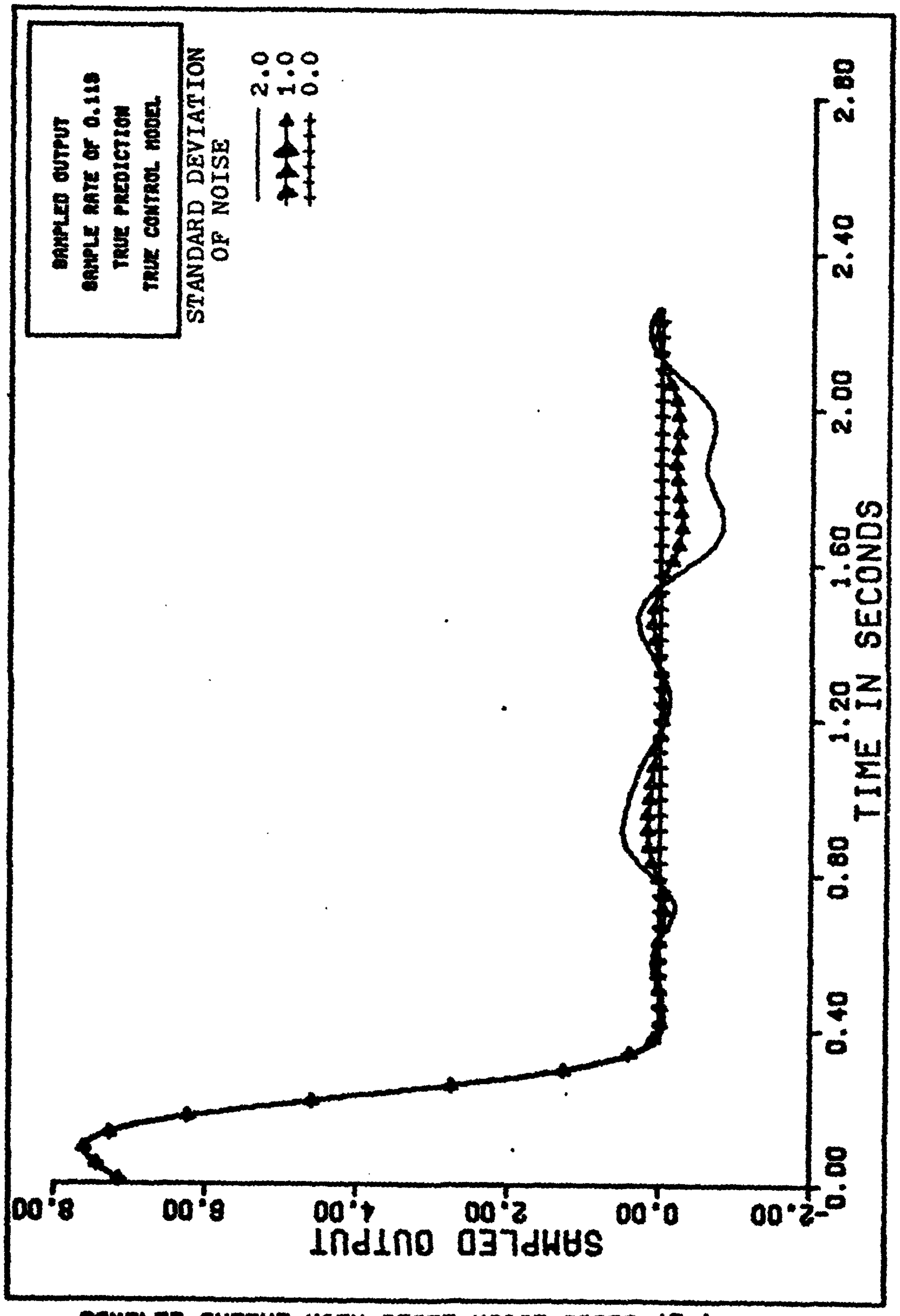
Fig. 20




CONTROL INPUT WITH INPUT NOISE ADDED AT A SAMPLE RATE OF .113 SECONDS



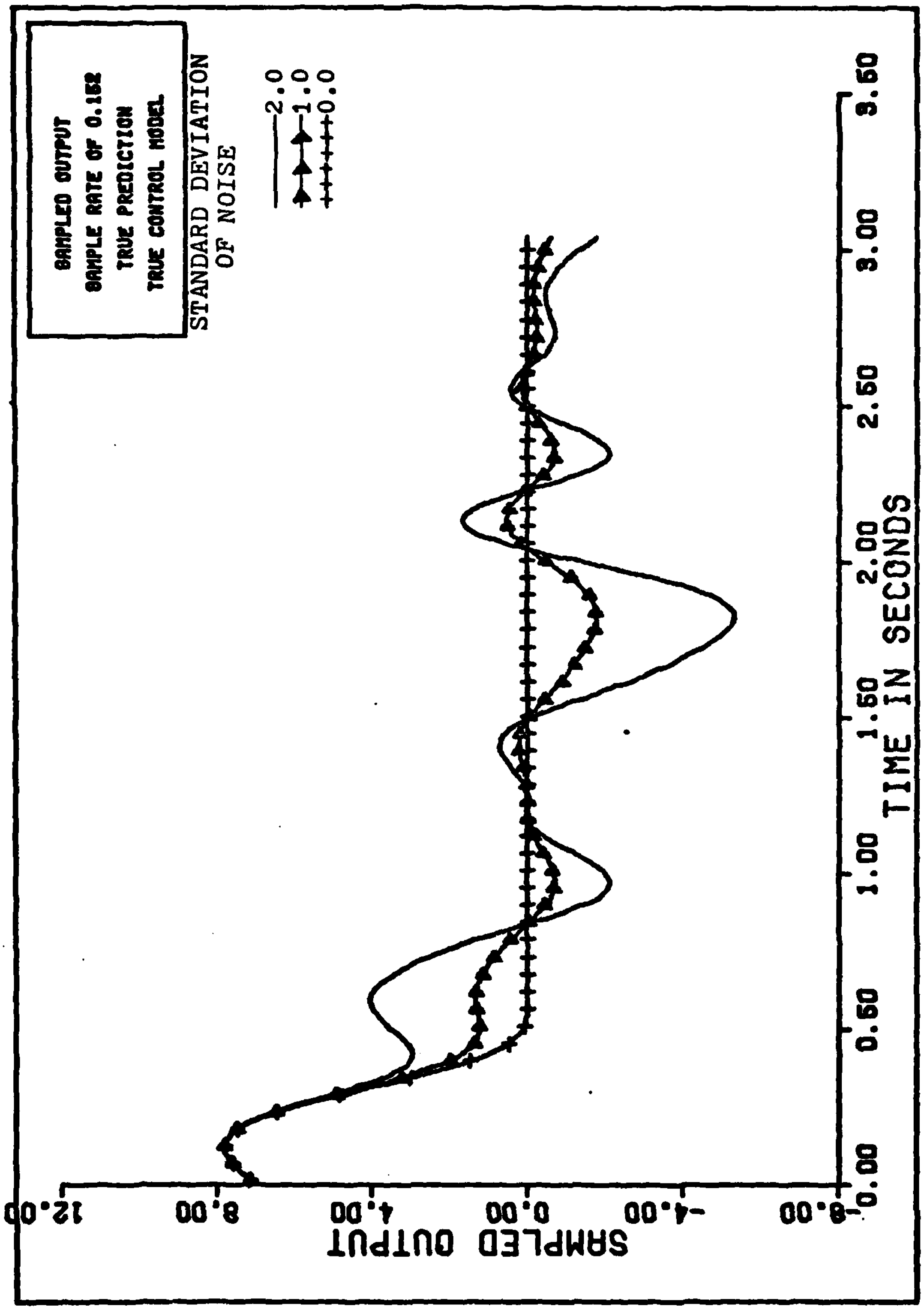
Fig. 21




SAMPLED OUTPUT WITH STATE NOISE ADDED AT A
SAMPLE RATE OF .113 SECONDS



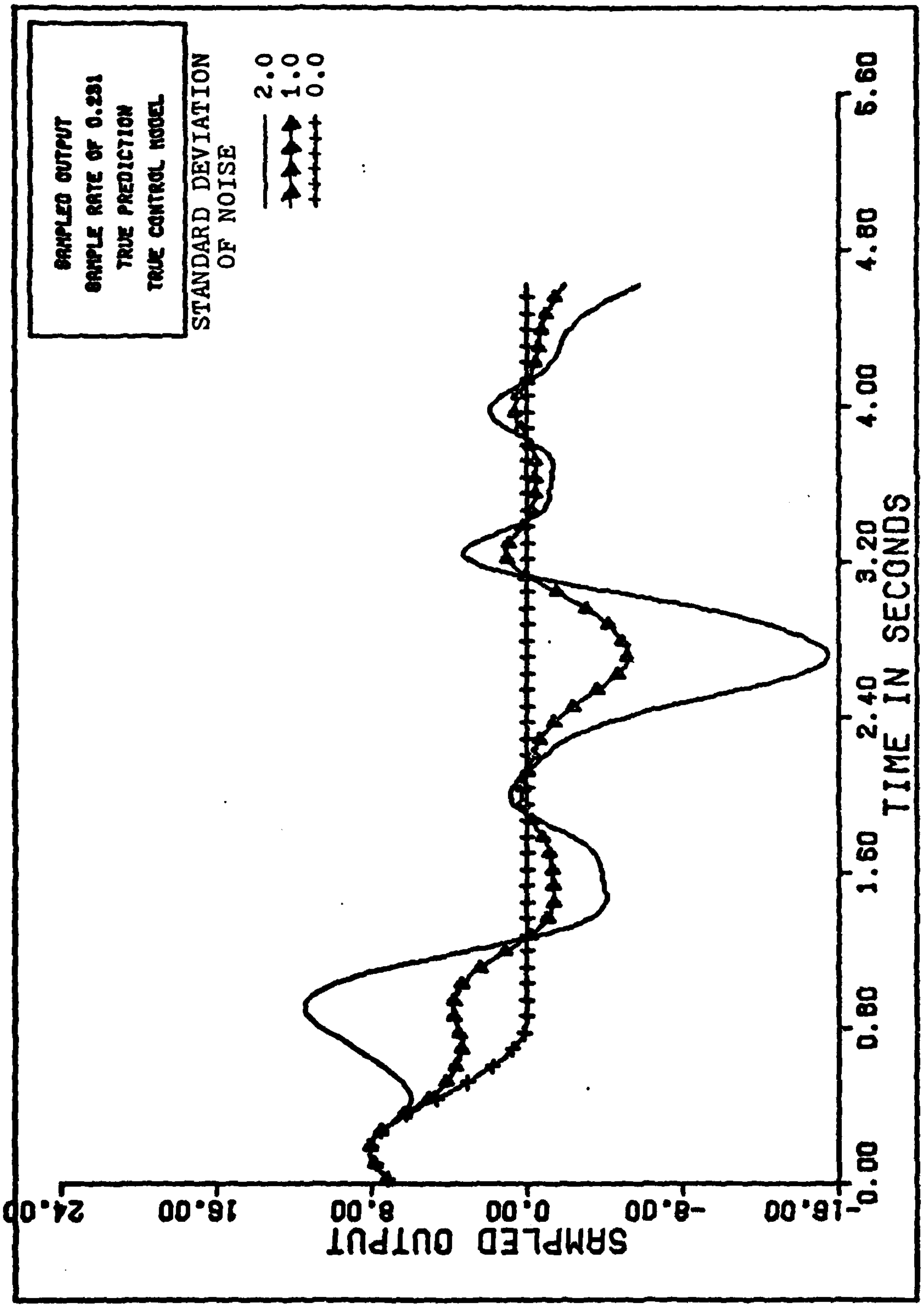
Fig. 22




SAMPLED OUTPUT WITH STATE NOISE ADDED AT A SAMPLE RATE OF .152 SECONDS



Fig.23




SAMPLED OUTPUT WITH STATE NOISE ADDED AT A SAMPLE RATE OF .231 SECONDS



Fig. 24

TABLE 5

Output Responses With Noise Corruption of the States

| CONTROL SAMPLE TIME | OUTPUT RESPONSE WITH STATE NOISE STRENGTHS OF | | | | 1/K |
|---|---|---|---|---|---|
| | .5 | 1.0 | 1.5 | 2.0 | |
| .04 | CS | CS | CS | CS | .000416 |
| .085 | S | CS | CS | CS | .00925 |
| .113 | S | S | CS | CS | .10664 |
| .152 | CS | CS | CS | CS | .32356 |
| .182 | CS | CS | CS | CS | .10714 |
| .214 | CS | CS | CS | US | .0093 |
| .231 | CS | CS | CS | US | .10646 |
| .5 | CS | CS | US | US | .00931 |
| .623 | CS | US | US | US | .0000355 |
| .835 | CS | CS | CS | US | .000423 |
| .954 | CS | CS | CS | CS | .09551 |

S - STABLE

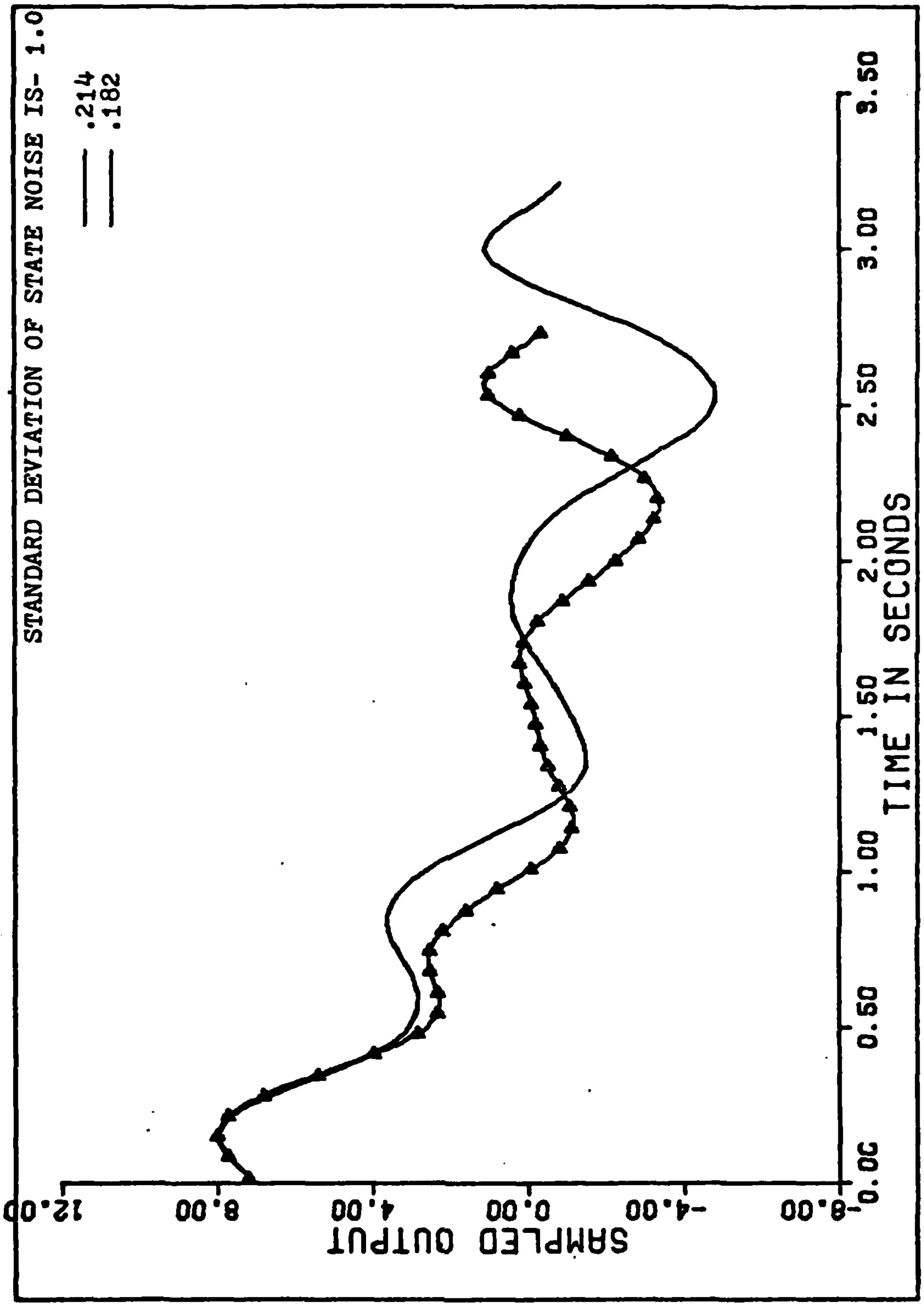
CS - CONDITIONALLY STABLE

US - UNSTABLE

From Table 5, one notices that the stability of the output, when noise is added to the states, is a function of sample rate. With faster sample rates, the system is able to correct the error induced by the state noise. The condition number is also a factor as seen in Fig 25. The output associated with the "poorer conditioned" sample rate (.214) is not as good as the output at the sample of .182. To best combat state noise corruption, one would like to use OPDEC at a fast sample rate that has a relatively good condition value. For the model used, a sample rate of .113 seems to have the best results (Fig 22).

<u>Input and State Noise Added</u>

This section concerns itself with the addition of both input and state noise. Looking at Tables 4 and 5, and trying to anticipate which




SAMPLED OUTPUT WITH STATE NOISE ADDED AT
SAMPLE RATES OF .214 and .182 SECONDS



Fig. 25

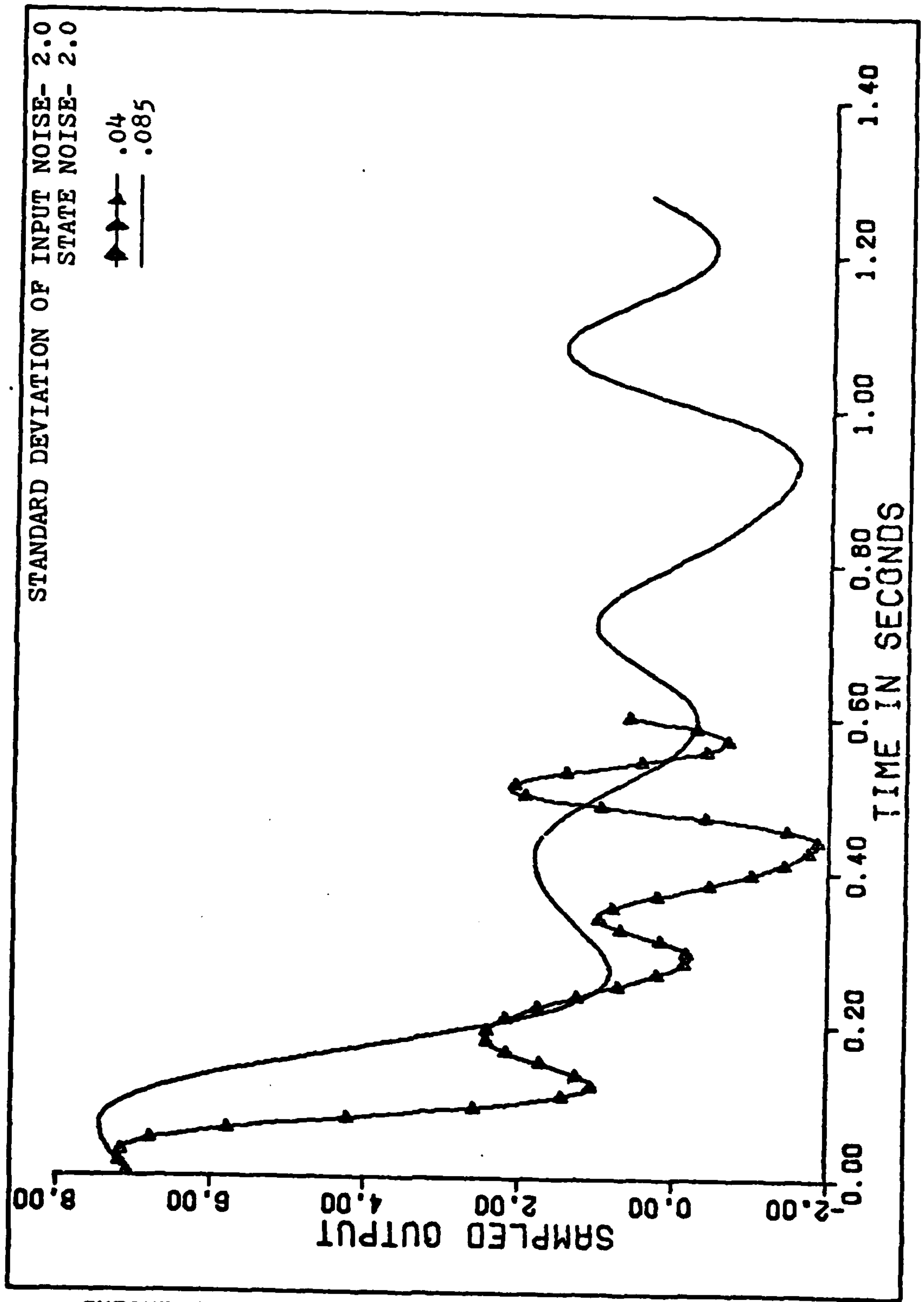
sample rate would give the best output results when both input and state noise is added, one would probably choose the following sample rates of .04, .085, .113, .152, .182 and .231. After a multitude of computer runs, implementing OPDEC with both noises added and at all the selected sample rates, the expected sample rates show the best robustness. Figures 26, 27, and 28 show the sampled output with both input and state noise added, and as one expects, .113 is the most robust.

This section has investigated the best way to combat noise in a system when using OPDEC for control. The conclusion appears to be, to run the system at a fast sample rate that is also well conditioned. The system seems to be able to correct the induced errors caused by noise when it is at a faster sample rate. In effect the errors can be "negated" before they significantly degrade the system response when operating at the faster sample rate.

## Effects of Model Mismatch

The program that implements OPDEC also has the option to use a perturbed model. This perturbed model can be used in either the prediction phase, control phase or both phases simultaneously. A mathmatical model can only approximate a system's response, but this mathmatical model is used to try and control the system. Since the model is in error the controller must be able to cope with this error without causing the system to go unstable. The perturbed model analysis in this section tries to simulate these model mismatching errors. Using this perturbed model analysis enables a person to see which phase (prediction or control) is the most sensitive to model errors.

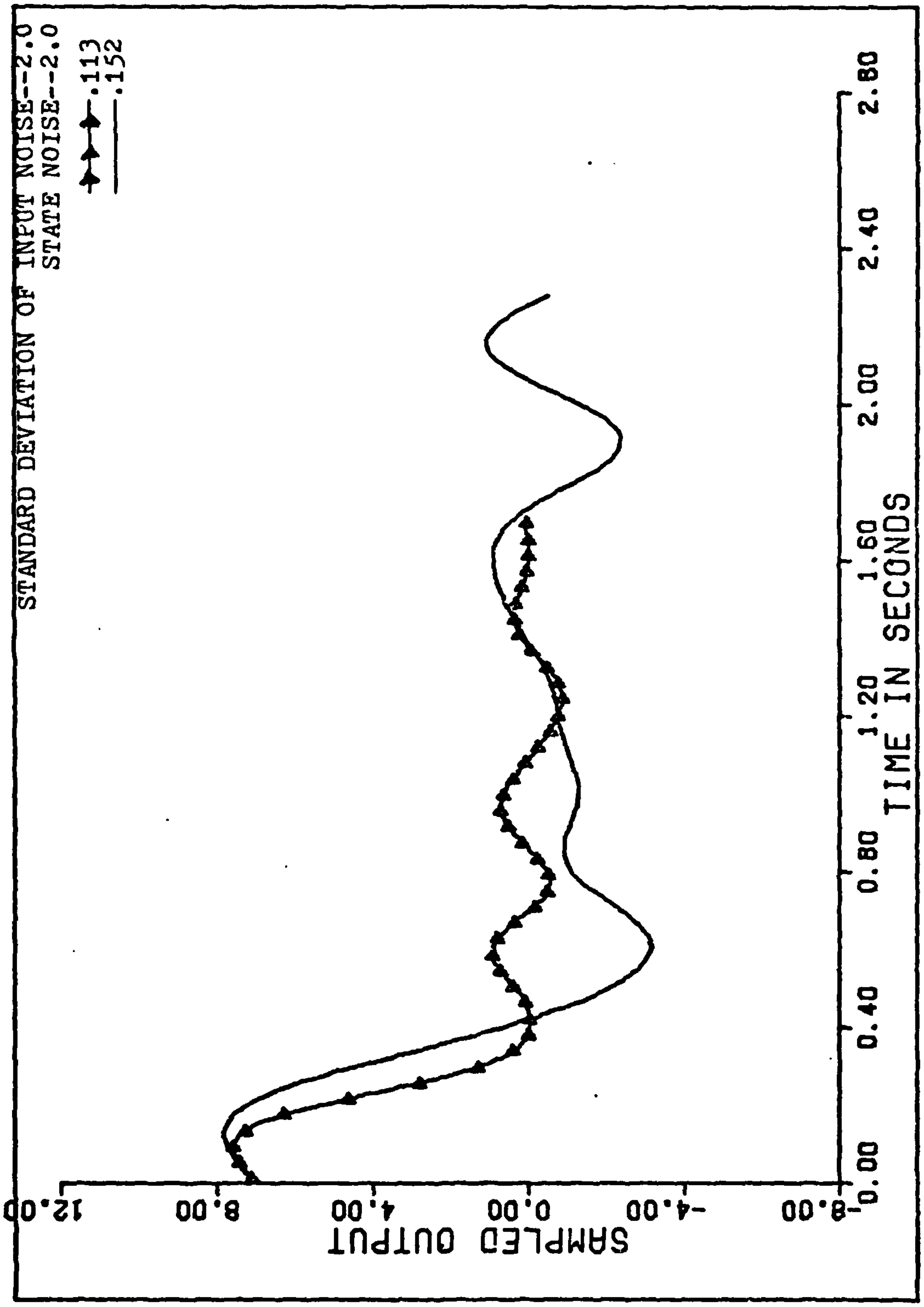
To generate the "perturbed model" the basic fourth order model eigenvalues are shifted in the model versus the "true system". The reasons for perturbing the eigenvalues instead of the Hankel matrix




OUTPUT WITH INPUT AND STATE NOISE ADDED AT SAMPLE RATES OF .04 and .085 SECONDS

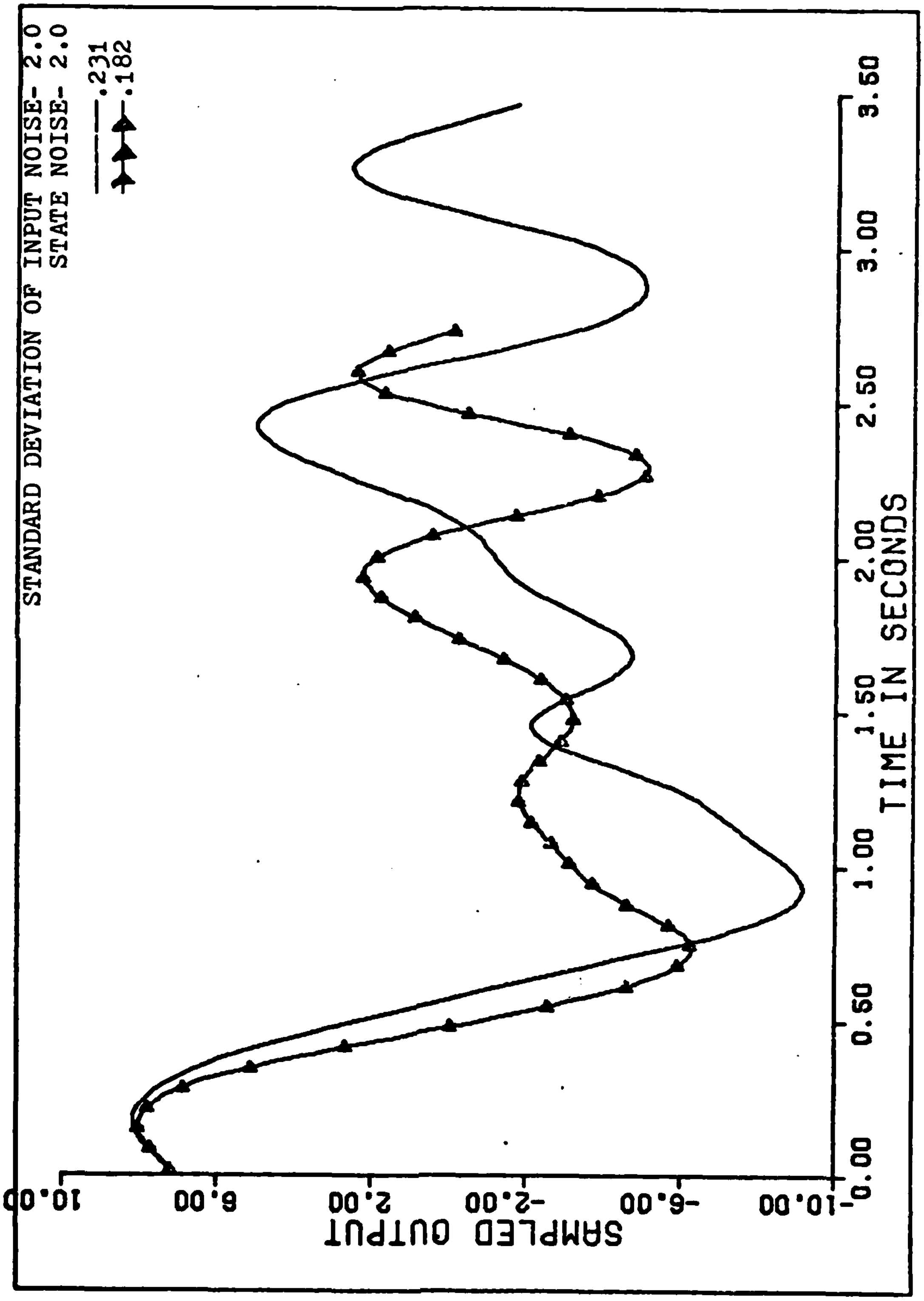


Fig. 26




OUTPUT WITH INPUT AND STATE NOISE ADDED AT SAMPLE RATES OF .113 and .152 SECONDS

Fig. 27




OUTPUT WITH INPUT AND STATE NOISE ADDED AT
SAMPLE RATES OF .231 and .182 SECONDS



Fig. 28

perturbations are not well understood. These perturbed eigenvalues are then used to create the perturbed model. Specifically, two perturbed models are used in the section to demonstrate robustness characteristics of OPDEC. The transfer function for the two perturbed systems are:

10% perturbation

$$\frac{+\ 8474}{S^4+1.66S^3+314.1S^2+360.6S+8474}$$

EIGENVALUES

$-.605 + j\ 5.4$
$-.605 + j\ 5.4$
$-.225 + j\ 16.94$
$-.225 + j\ 16.94$

15% perturbation

$$\frac{+\ 8285}{S^4+1.69S^3+340.6S^2+408S+8285}$$

EIGENVALUES

$-.6325 + j\ 5.1$
$-.6325 - j\ 5.1$
$-.2125 + j\ 17.71$
$-.2125 - j\ 17.71$

OPDEC's program was then implemented at all the selected sample rates, using the perturbed model in the following fashion:

1. The perturbed model was used in the prediction phase and the true model was used in the control phase. (False prediction, true control)

2. The true model was used in the prediction phase and the perturbed model was used in the control phase. (True prediction, false control)

3. The perturbed model was used in both prediction and control phase. (False prediction, false control)

The sampled output responses, for all selected sample rates, were stable.

According to Reid (Ref 3) the norm of the difference between the true Hankel matrix and the error Hankel matrix ($E = H_t - H_e$) gives an indication of the stability of the output. Tables 7 and 8 show the selected sample rates, the condition number, the norm of the true Hankel matrix $||H_t||$, the norm of the difference between the true Hankel matrix and the error Hankel matrix, $||E||$, the norm of the E matrix divided by the norm of the true Hankel matrix, and the condition number times this value for the 10% and 15% error models used.

TABLE 6

10% Perturbed Model

| DELT | K | $\|\|H_t\|\|$ | $\|\|E\|\|$ | $\|\|E\|\|/\|\|H_t\|\|$ | $K \cdot \|\|E\|\|/\|\|H_t\|\|$ |
|---|---|---|---|---|---|
| .040 | 2403.555 | .652 | .078 | .120 | 289.085 |
| .085 | 108.102 | 1.562 | .401 | .257 | 27.754 |
| .113 | 9.055 | 1.566 | .553 | .353 | 3.197 |
| .152 | 3.091 | 1.718 | .559 | .325 | 1.006 |
| .182 | 9.334 | 2.086 | .937 | .449 | 4.191 |
| .214 | 107.511 | 2.366 | 1.121 | .474 | 50.957 |
| .231 | 9.393 | 2.406 | .874 | .363 | 3.412 |
| .500 | 107.367 | 4.001 | 2.543 | .635 | 68.227 |
| .623 | 28151.136 | 2.012 | 2.941 | 1.462 | 41155.396 |
| .835 | 2363.574 | 1.350 | .911 | .675 | 1594.370 |
| .954 | 10.470 | .836 | .682 | .816 | 8.543 |

TABLE 7

15% Perturbed Model

| DELT | K | $\|\|H_t\|\|$ | $\|\|E\|\|$ | $\|\|E\|\|/\|\|H_t\|\|$ | $K \cdot \|\|E\|\|/\|\|H_t\|\|$ |
|---|---|---|---|---|---|
| .040 | 2403.555 | .625 | .123 | .189 | 454.633 |
| .085 | 108.102 | 1.562 | .566 | .362 | 39.147 |
| .113 | 9.055 | 1.566 | .750 | .479 | 4.336 |
| .152 | 3.091 | 1.718 | .750 | .436 | 1.349 |
| .182 | 9.334 | 2.086 | 1.226 | .588 | 5.486 |
| .214 | 107.511 | 2.366 | 1.287 | .544 | 58.498 |
| .231 | 9.393 | 2.406 | 1.288 | .535 | 5.026 |
| .500 | 107.367 | 4.001 | 3.589 | .897 | 96.315 |
| .623 | 28151.136 | 2.012 | 3.141 | 1.561 | 43943.039 |
| .835 | 2363.574 | 1.350 | 1.104 | .817 | 1932.136 |
| .954 | 10.470 | .836 | .903 | 1.081 | 11.314 |

Looking at the tables, at the same sample rate the norm of E increases as the amount of perturbation increases. If the norm of E is less than one, the output response is suppose to be stable. This section will try and verify this theoretical result by looking at the output response at all of the sample rates selected using the perturbed models and seeing if the output is stable or unstable. Table 8 shows the effects which the 10% perturbed model has on the sampled output.

TABLE 8

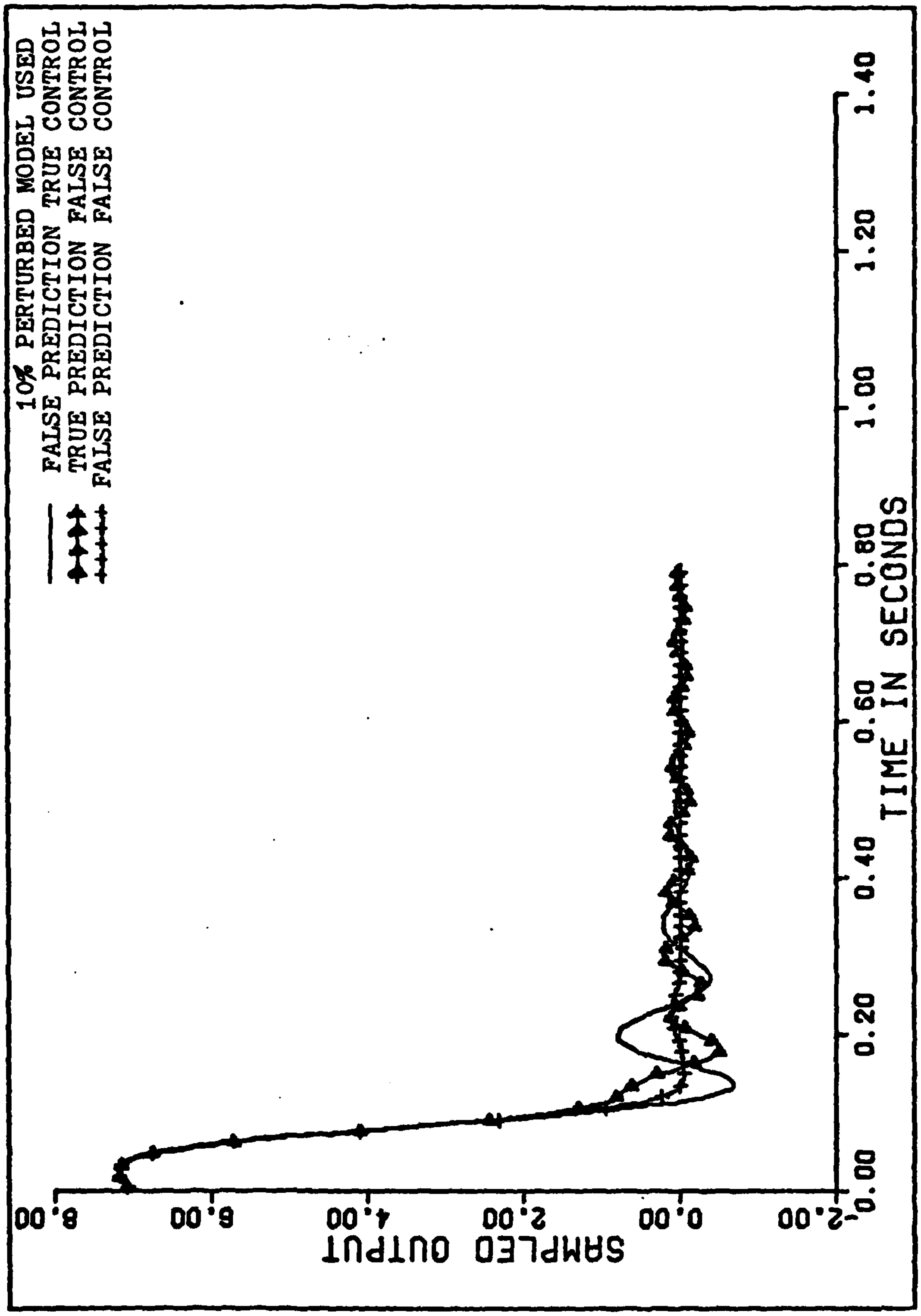
Output Responses With the 10% Perturbed Model

| SAMPLE RATE | OUTPUT RESPONSE WITH TRUE PREDICTION FALSE CONTROL | FALSE PREDICTION TRUE CONTROL | FALSE PREDICTION FALSE CONTROL |
|---|---|---|---|
| .04 | S | S | S |
| .085 | S | S | S |
| .113 | S* | S* | S |
| .152 | S | S* | S |
| .182 | S* | US | S* |
| .214 | US | US | US |
| .231 | US | US | S |
| .5 | US | US | US |
| .623 | US | US | US |
| .835 | US | US | US |
| .954 | US | US | S |

* The time needed to reach the final value was greater than 20 times the sample rate used.

The output responses using the 10% perturbed model, for sample rates of .04, .085, .113, and .152 are in Figs 29, 30, 31 and 32 respectively. From these figures one can see that the prediction phase is more sensitive than the control phase to model error. But what is really interesting is the output response is better when the error model is used in both prediction and control phases.

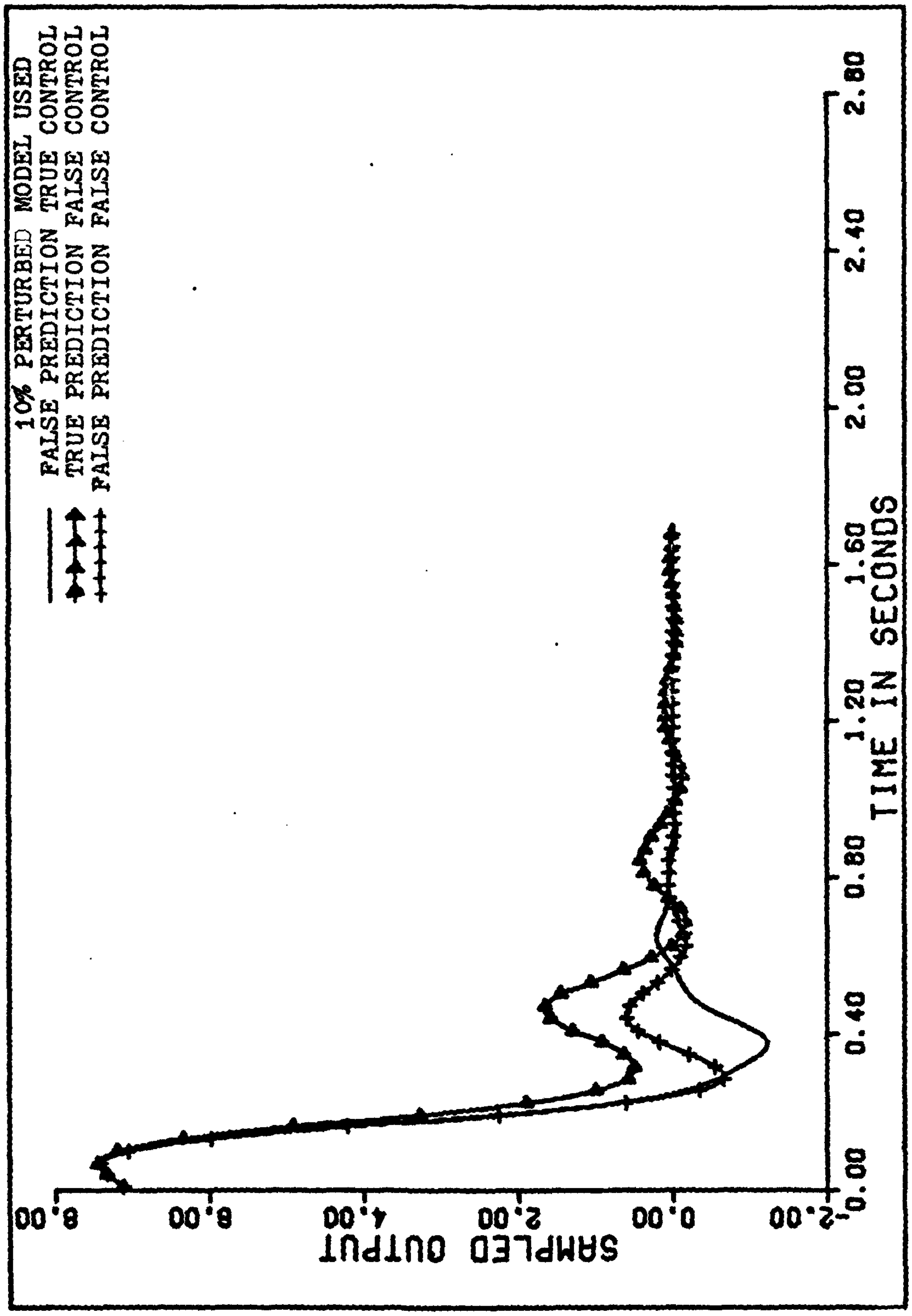
Table 9 shows the effects the 15% perturbed model has on the sampled output.




**SAMPLED OUTPUT WITH NO NOISE ADDED** USING THE
10% PERTURBED MODEL AT A SAMPLE RATE OF .04 SECONDS



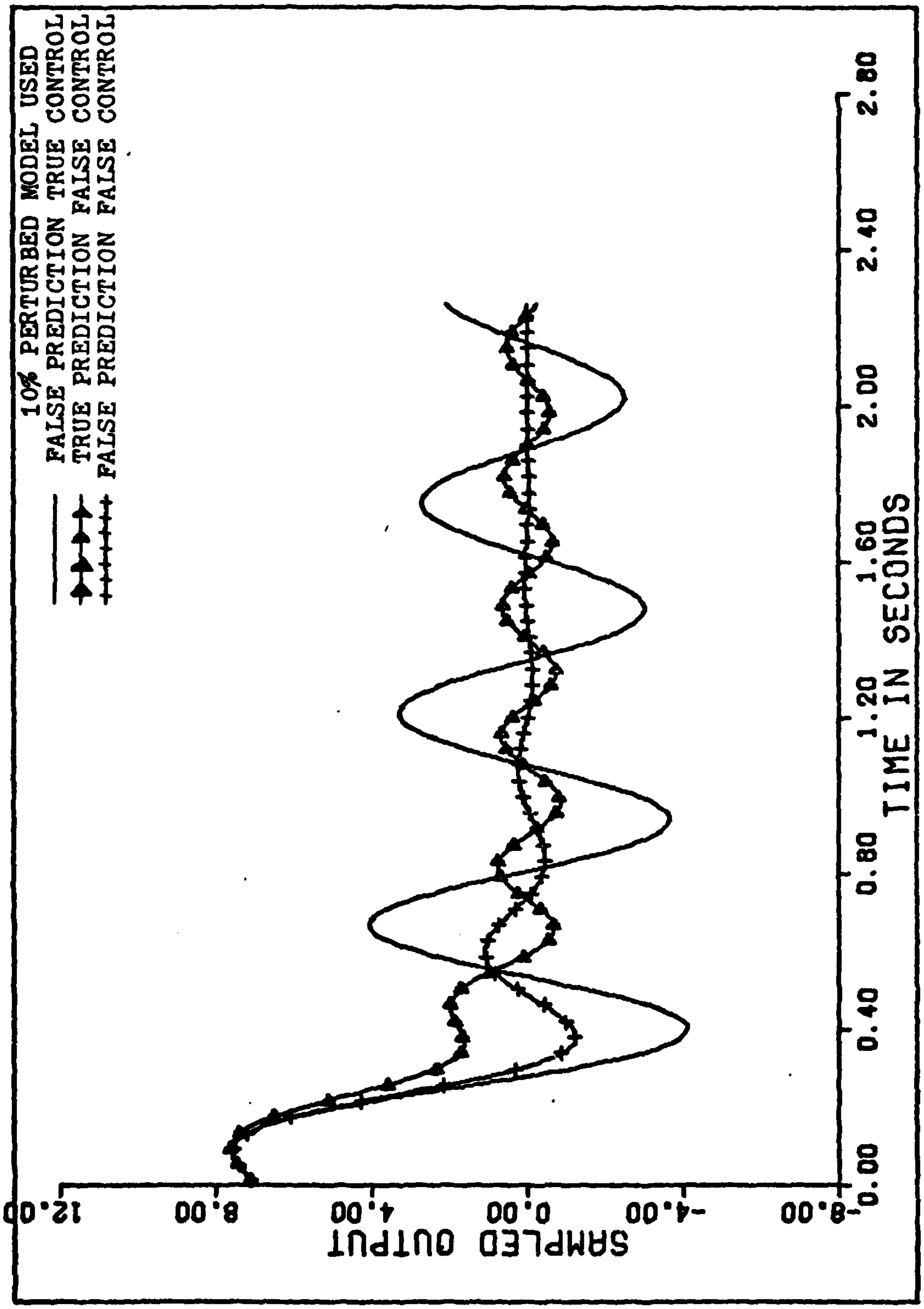
Fig. 29




SAMPLED OUTPUT WITH NO NOISE ADDED USING THE
10% PERTURBED MODEL AT A SAMPLE RATE OF .085 SECONDS

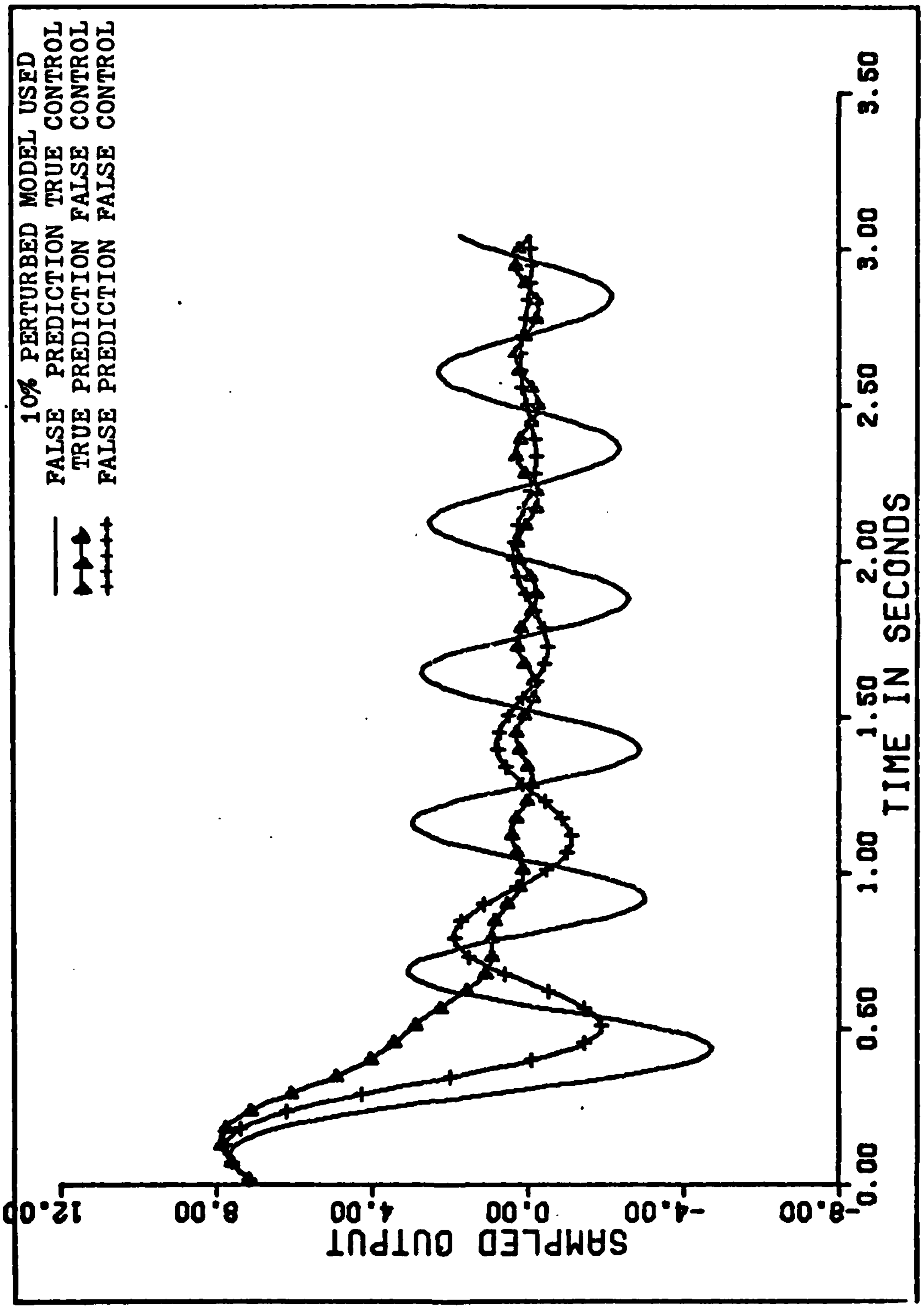


Fig. 30




SAMPLED OUTPUT WITH NO NOISE ADDED USING THE
10 % PERTURBED MODEL AT A SAMPLE RATE OF .113 SECONDS

Fig. 31




SAMPLED OUTPUT WITH NO NOISE ADDED USING THE
10% PERTURBED MODEL AT A SAMPLE RATE OF .152 SECONDS

Fig. 32

TABLE 9

Output Responses With 15% Perturbed Model

| SAMPLE RATE | OUTPUT RESPONSES WITH TRUE PREDICTION FALSE CONTROL | FALSE PREDICTION TRUE CONTROL | FALSE PREDICTION FALSE CONTROL |
|---|---|---|---|
| .04 | S | S | S |
| .085 | S | S | S |
| .113 | US | US | S |
| .152 | US | US | US |
| .182 | US | US | US |
| .214 | US | US | US |
| .231 | US | US | US |
| .5 | US | US | US |
| .623 | US | US | US |
| .835 | US | US | US |
| .954 | US | US | US |

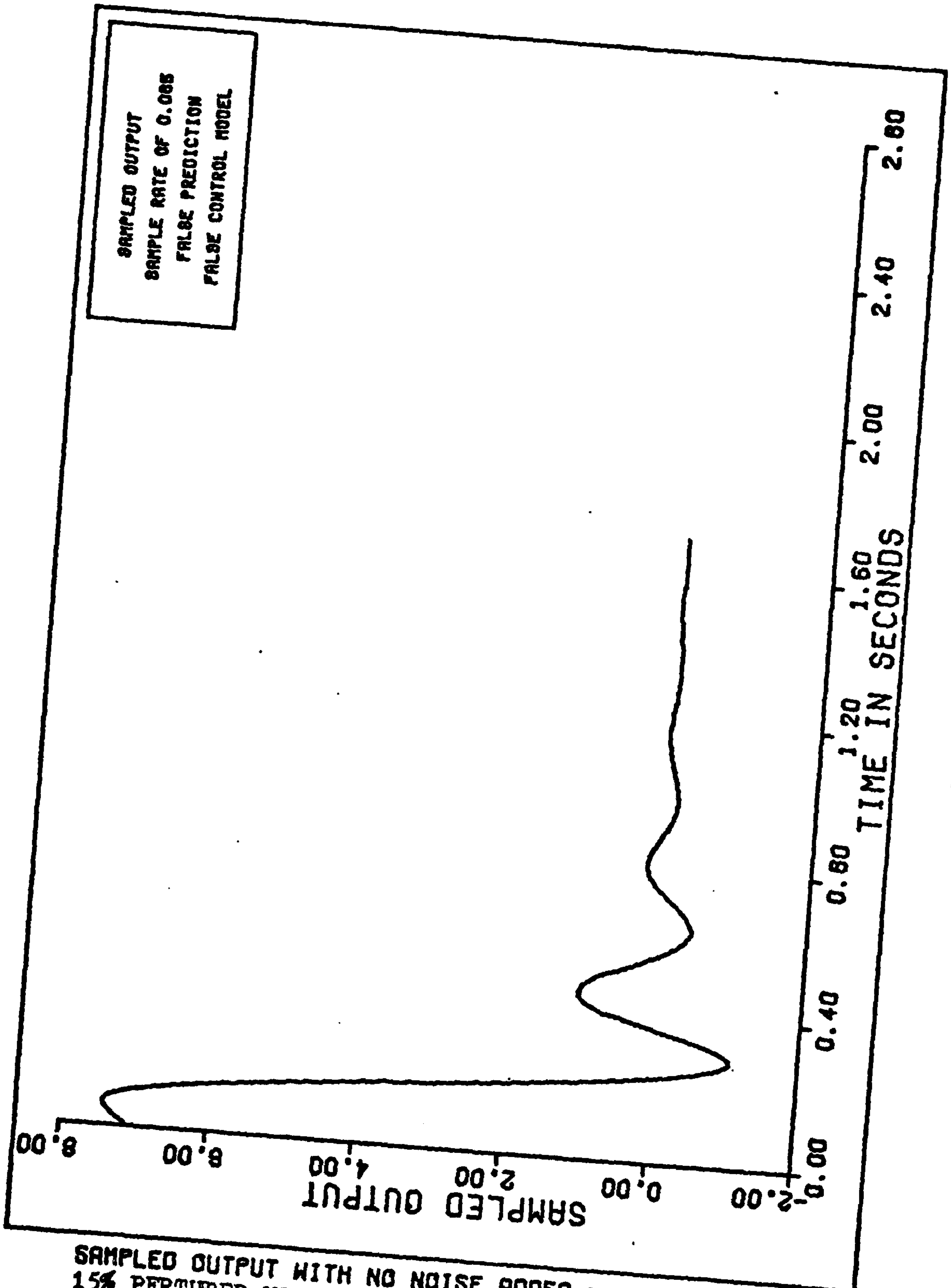
S - STABLE

US - UNSTABLE

The models errors are almost too great to overcome when the perturbation is 15%. The output responses using the 15% perturbed model for sample rates of .085 and .113 are in Fig 33 and 34. These sample rates, as predicted by Table 7, are stable. It seems that the norm of the difference between the two Hankel matrix and the error Hankel matrix if less than one indicates stability.

<u>Combination of Noise and Model Errors</u>

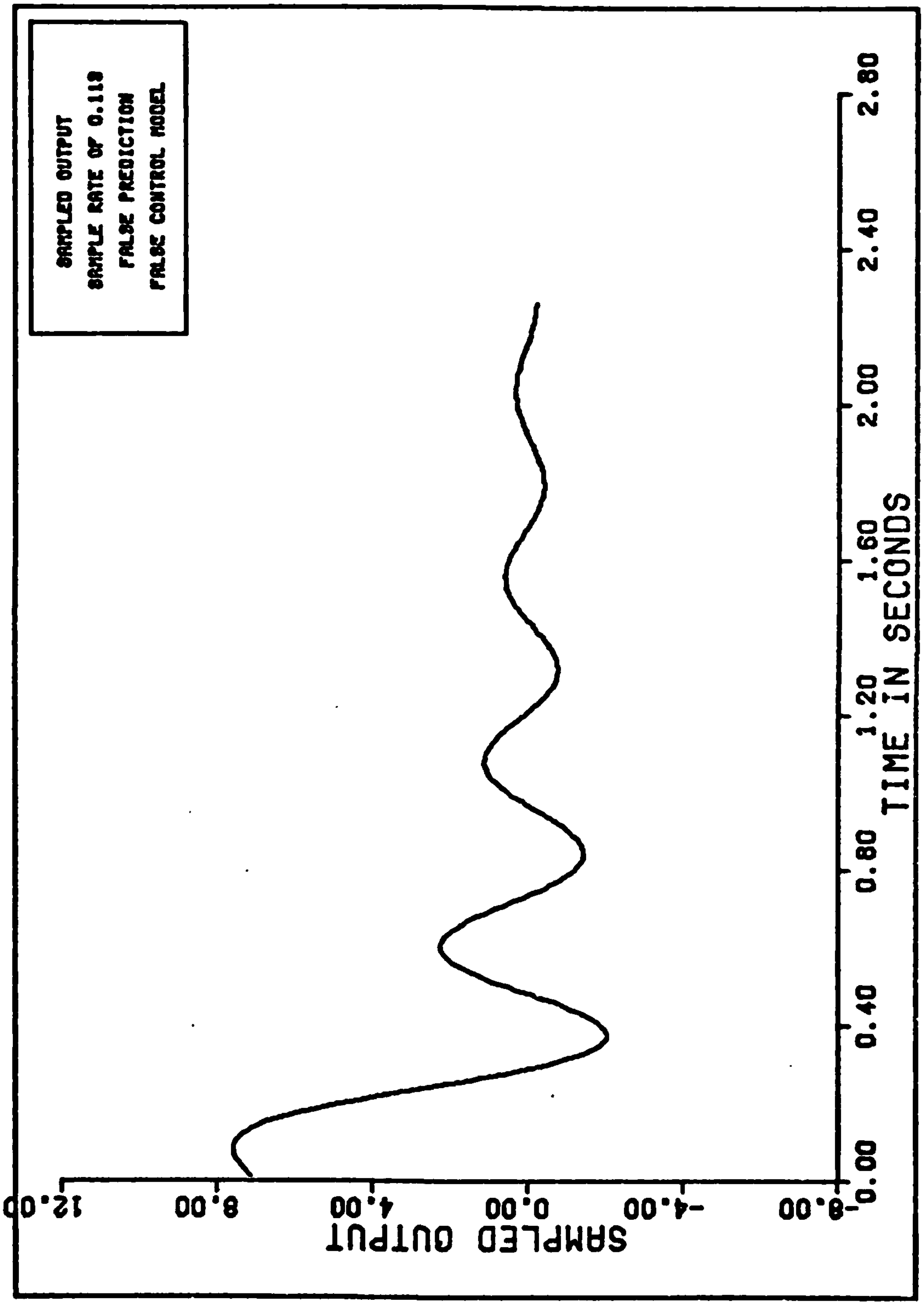
This section investigates the efforts of input and state noise when erroneous models are used to control the system. OPDEC's program was then implemented with the perturbed model used in both prediction and control phases with just input noise, then just state noise, and finally both input and state noise. The justification for implementing OPDEC in this fashion is that one will not be using different models for




SAMPLED OUTPUT WITH NO NOISE ADDED USING THE 15% PERTUBED MODEL AT A SAMPLE RATE OF .085 SECONDS



Fig. 33




SAMPLED OUTPUT WITH NO NOISE ADDED USING THE
15% PERTURBED MODEL AT A SAMPLE RATE OF .113 SECONDS



Fig. 34

the production phase as compared to the control phase. The same model will be used in both phases.

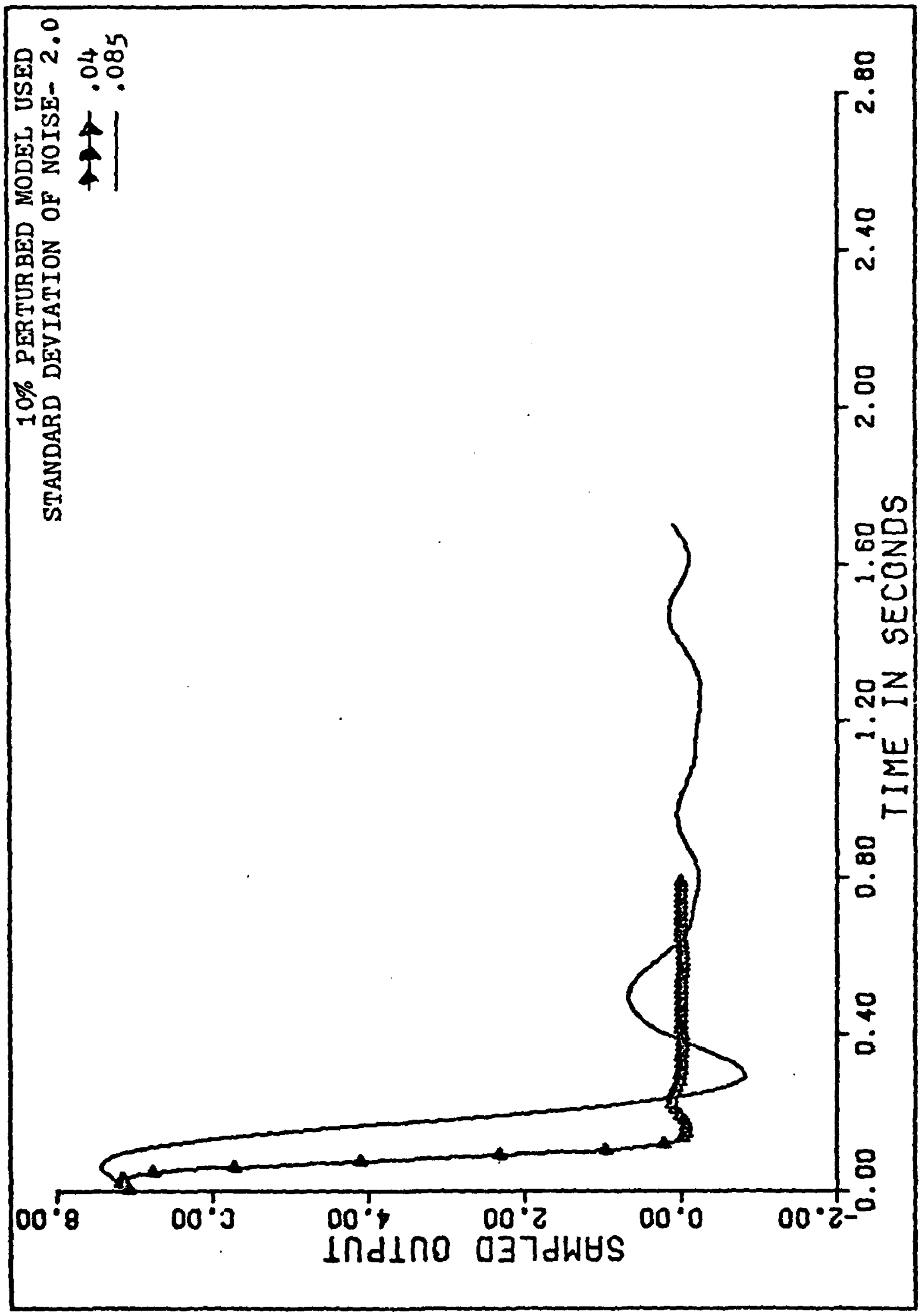
OPDEC's program was implemented using the 10% perturbed model, input noise was added first, then state noise, then the combination of the two noises. This was done for all the sample rates selected, except the sample rates which by Table 4 were unstable. Shown in Figures 35 and 36 is the sampled outputs using the 10% perturbed model with input noise added. The sampled outputs with state noise added is shown in Figures 37 and 38. The sampled outputs with both state and input noise added can be seen in Figures 39 and 40.

As expected, the faster sample rates which have larger controls are insensitive to input disturbances, but the addition of state noise has a more severe effect on the output. The severity is great enough to make the output response worse than the response at a sample rate of .113. When using a perturbed model, the influence of state noise becomes more severe. This is because of OPDEC's sensitivity in the prediction phase of its control law.

## Hankel Matrix Sensitivity Analysis

The use of OPDEC requires working with the Hankel matrix. Since the Hankel matrix is such a vital part of OPDEC, the analysis of the Hankel matrix under variations in the system pole/zero structure will give some indication of the overall robustness of OPDEC. Not only is the Hankel matrix an intrinsic part of control computation in OPDEC, it is also an important part in the selection of an "optimal" sample rate. The approach taken in this report is to see how sensitive the Hankel matrix is to pole and zero placement.

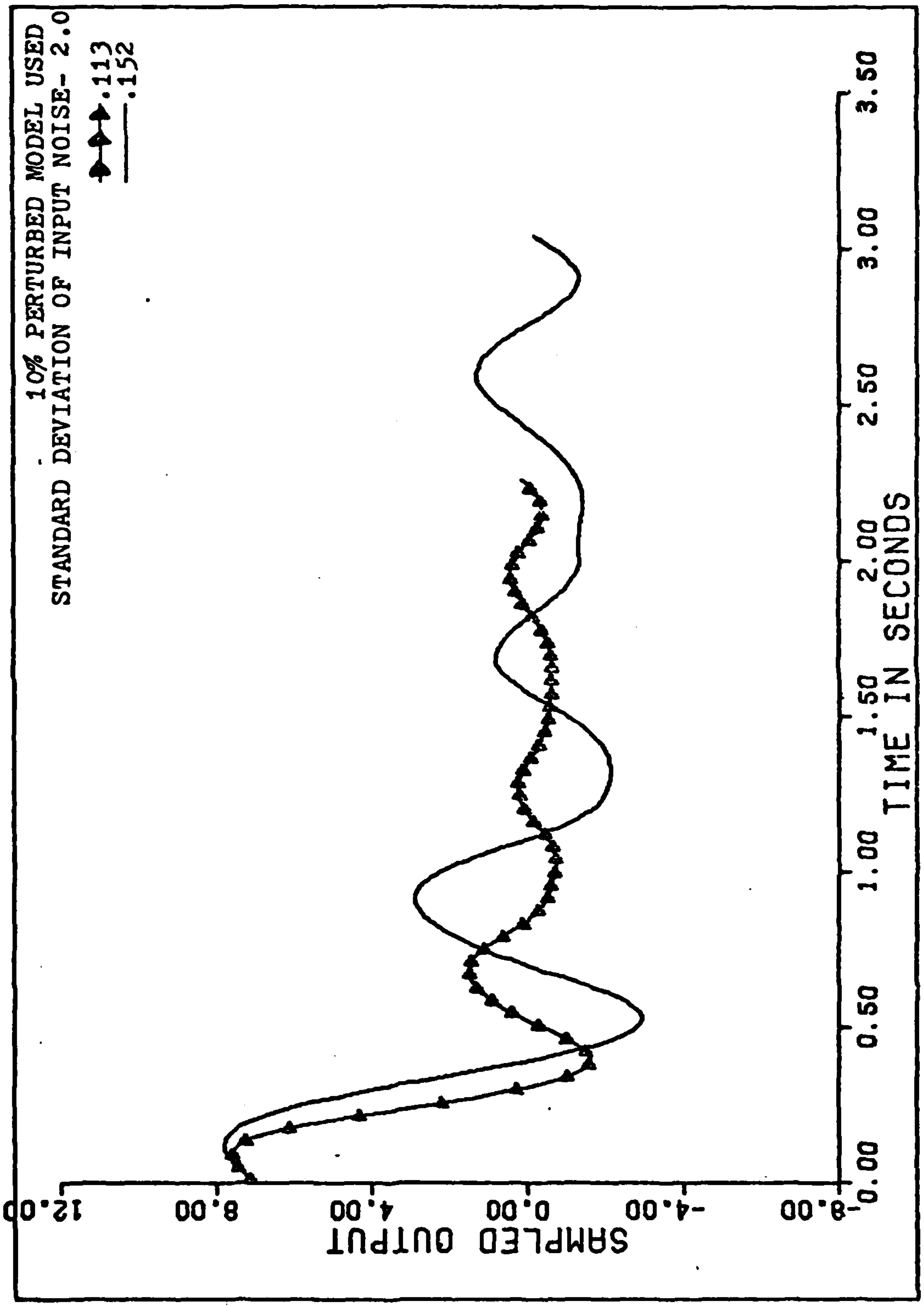
First the dominate pair of eigenvalues in the fourth order system




SAMPLED OUTPUT WITH INPUT NOISE ADDED USING THE 10% PERTURBED MODEL AT SAMPLE RATES OF .04 and .085 SECONDS

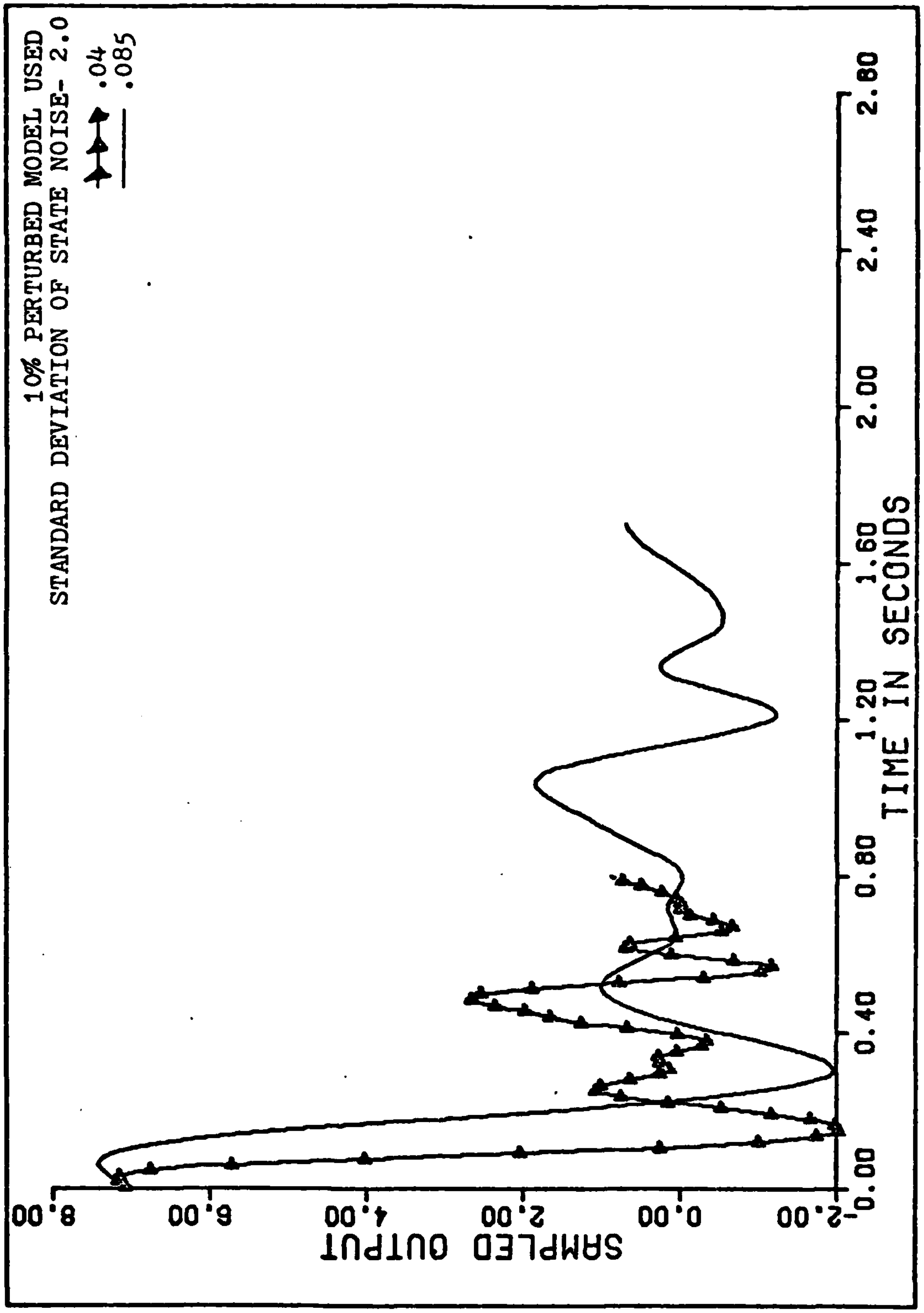


Fig. 35




SAMPLED OUTPUT WITH INPUT NOISE ADDED USING THE 10% PERTUBED MODEL AT SAMPLE RATES OF .113 and .152 SECONDS

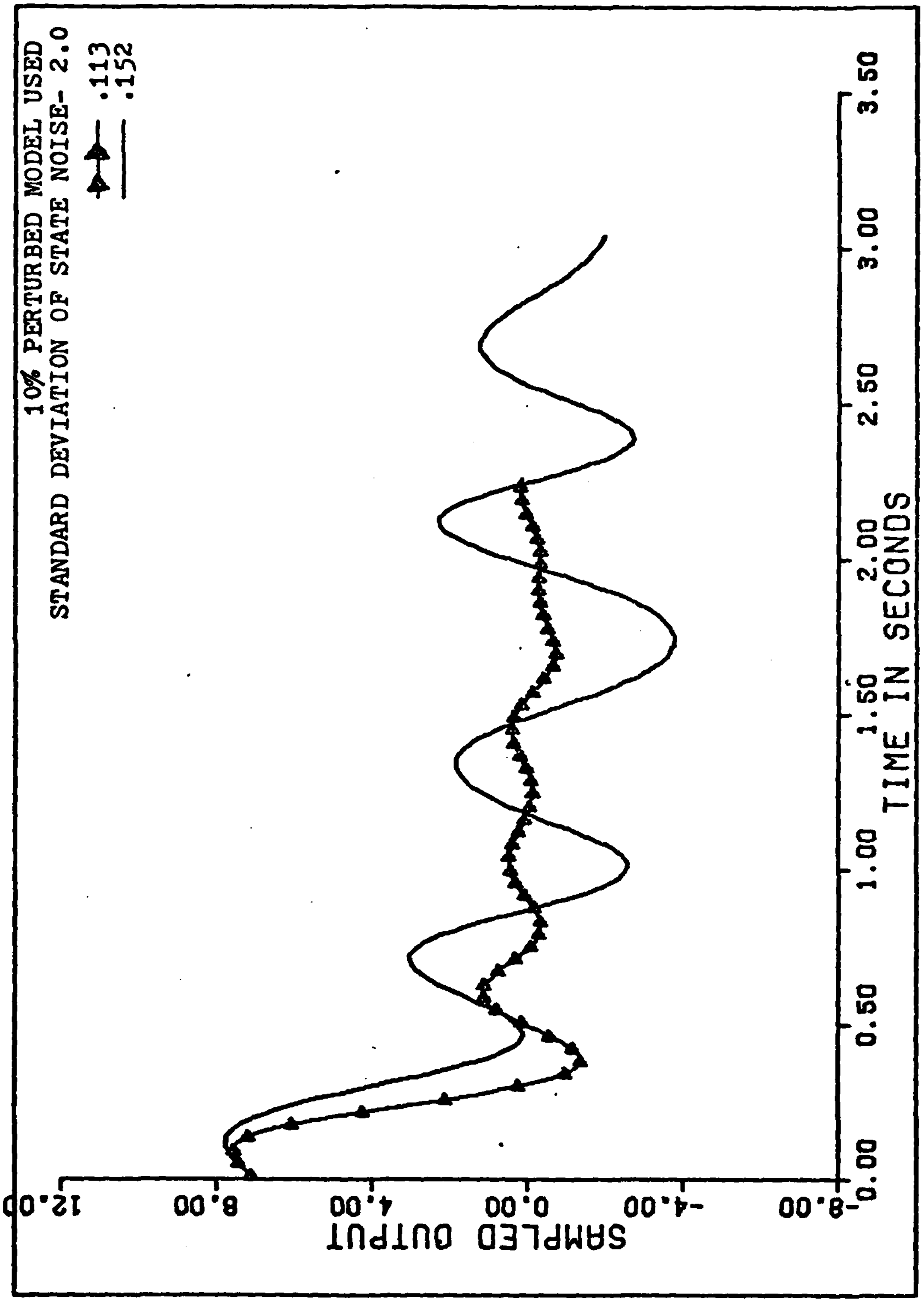
Fig. 36




SAMPLED OUTPUT WITH STATE NOISE ADDED USING THE 10% PERTURBED MODEL AT SAMPLE RATES OF .085 and .04 SECONDS.



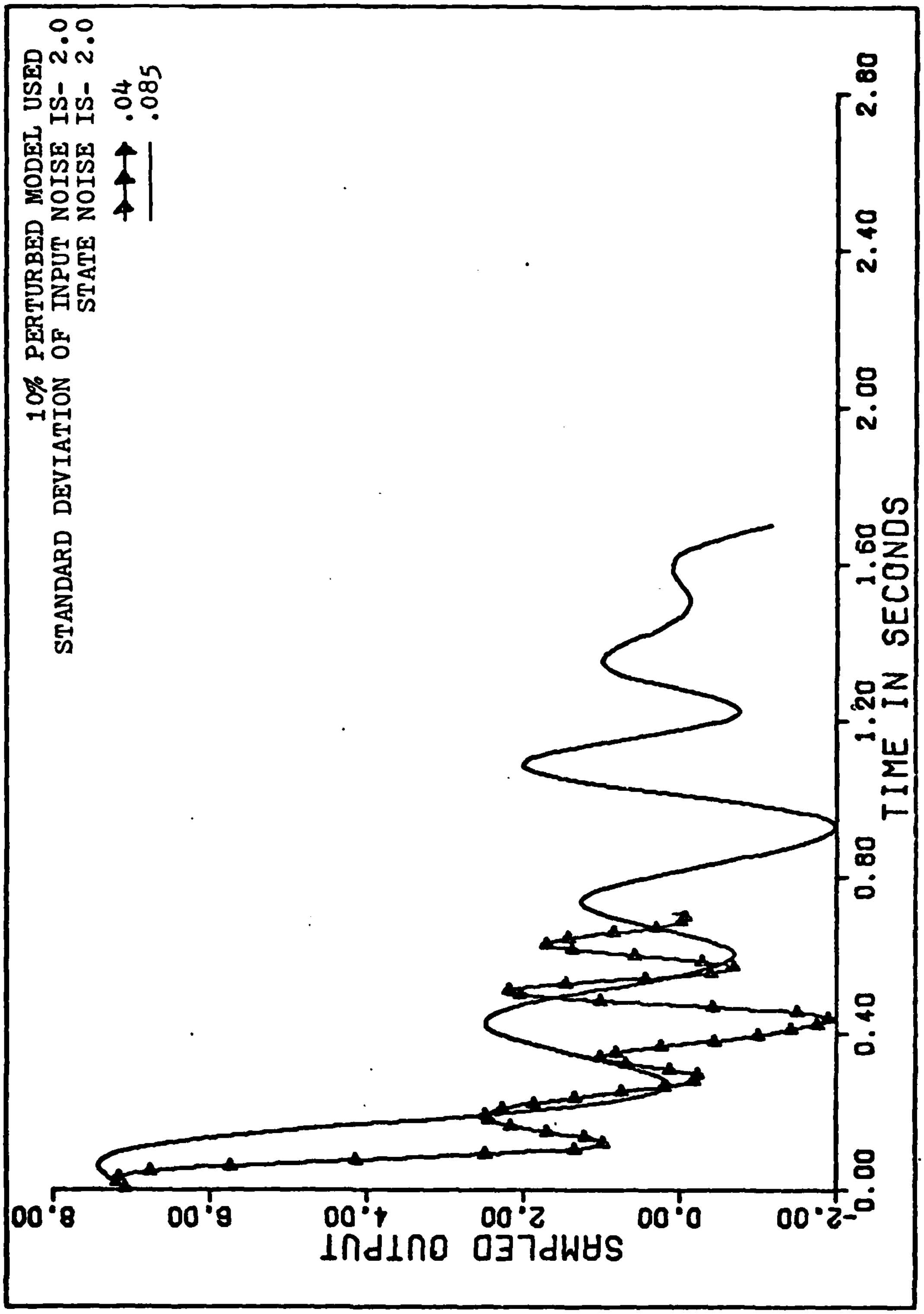
Fig. 37




SAMPLED OUTPUT WITH STATE NOISE ADDED USING THE 10% PERTURBED MODEL AT SAMPLE RATES OF .113 and .152 SECONDS.



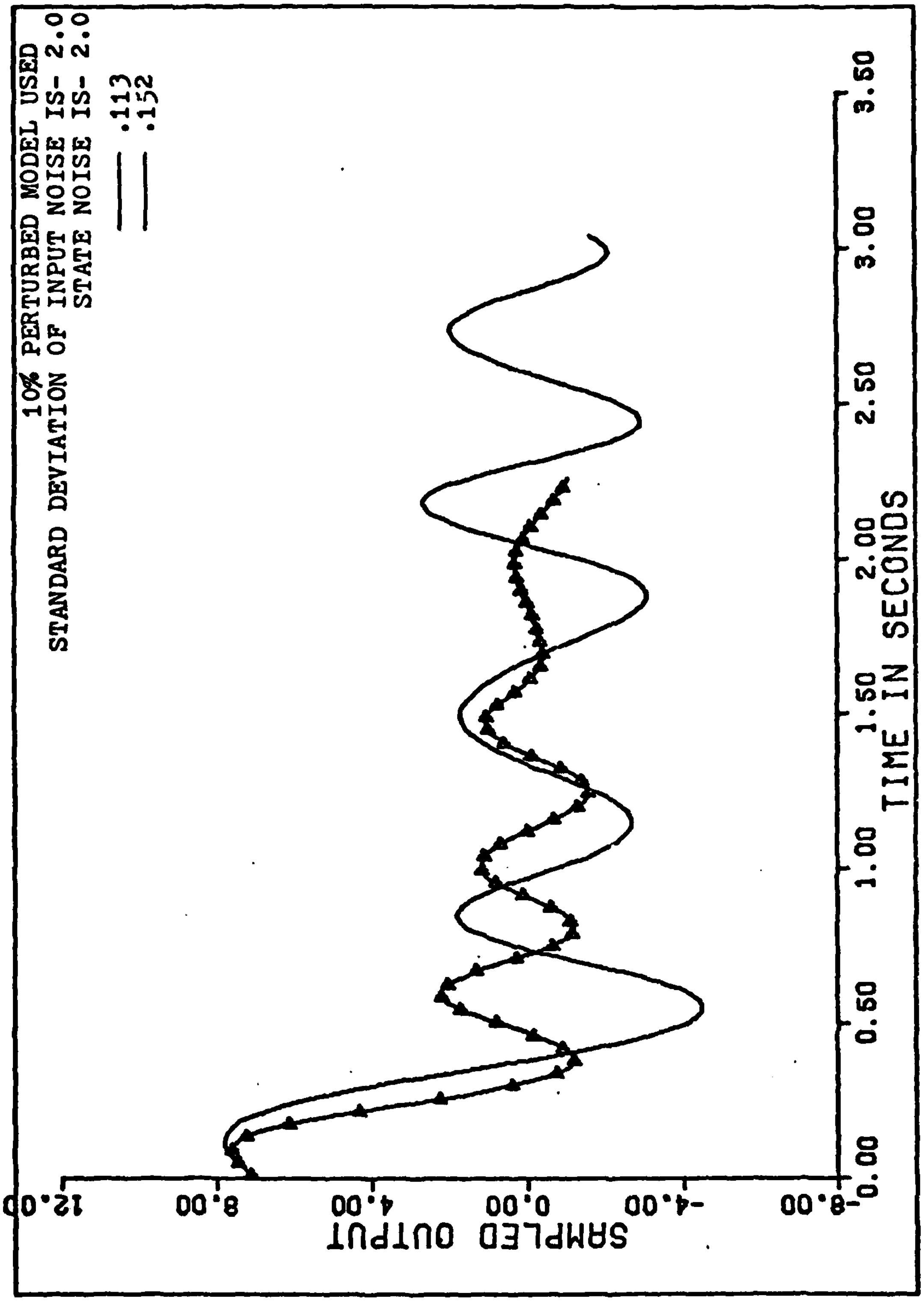
Fig. 38




OUTPUT WITH INPUT AND STATE NOISE ADDED USING THE 10% PERTURBED MODEL AT SAMPLE RATES OF .04 and .085 SECONDS.



Fig. 39




OUTPUT WITH INPUT AND STATE NOISE ADDED USING THE 10% PERTURBED MODEL AT SAMPLE RATES OF .113 and .152 SECONDS.



Fig. 40

$(-.25 \pm j\ 15.4)$ are perturbed slightly, then combined with the two other original eigenvalues to come up with a new perturbed system. This system is then used in the first FORTRAN program (Appendix A) to see the effects which pole placement has upon the condition number. The resulting condition number versus sample rate plot is then compared to the original system condition number plot. The original system condition number plot is again shown in Figure 41. Figure 42 and Table 10 show the perturbed eigenvalues used in the sensitivity analysis. Table 10 also has the optimal sample rate and one over condition number for each perturbed eigenvalue.

TABLE 10

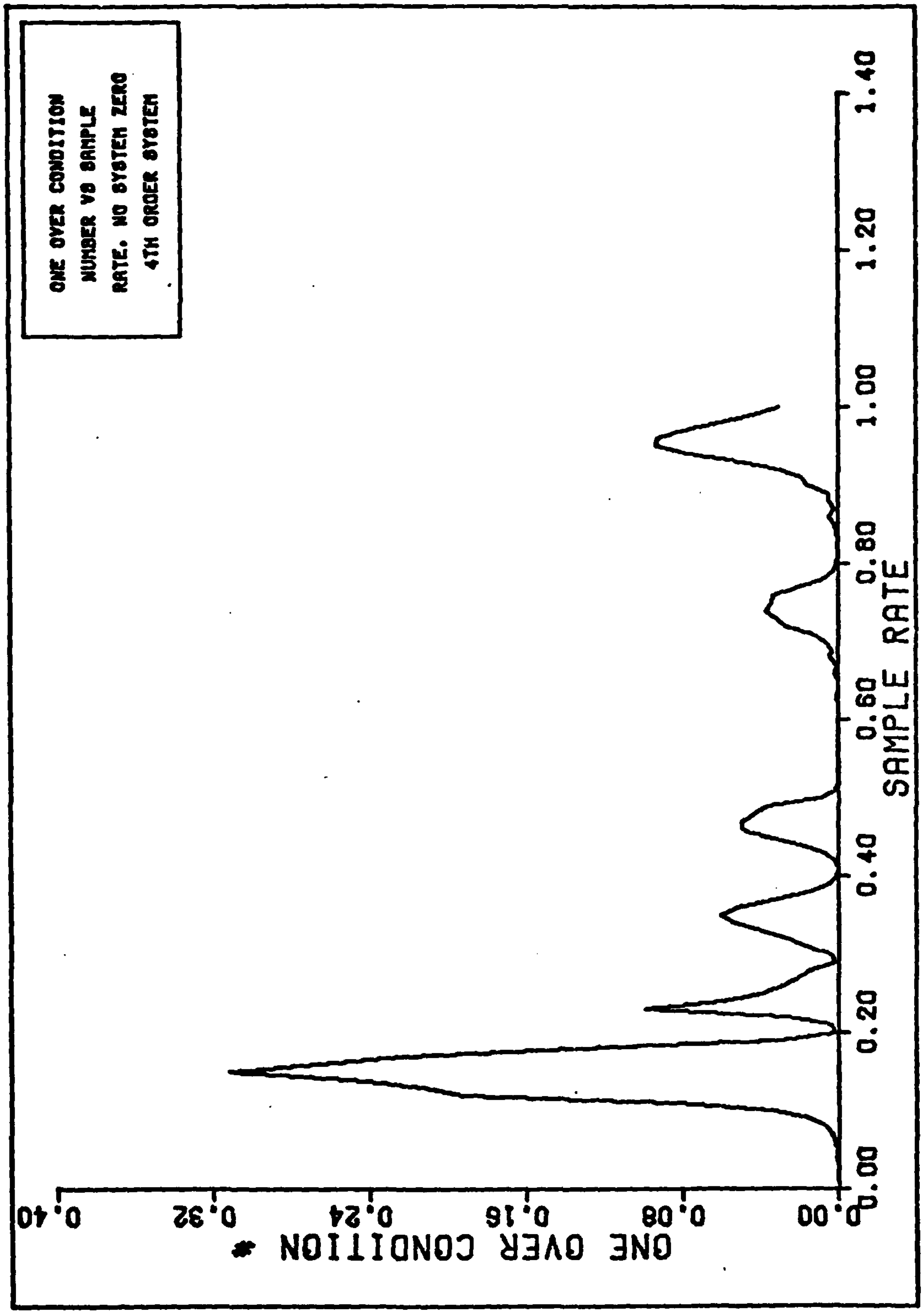
| PERTURBED EIGENVALUE | "OPTIMAL" SAMPLE RATE | MAXIMUM RECIPROCAL CONDITION NUMBER @ "OPTIMUM" SAMPLE RATE | Wn | DAMPING RATIO |
|---|---|---|---|---|
| $-.25 \pm j\ 15.4$! | .152 | .32356 | 15.402 | .016 |
| $-.25 \pm j\ 20$ | .118 | .23109 | 20.001 | .0125 |
| $-.25 \pm j\ 17.4$ | .136 | .2881 | 17.401 | .014 |
| $-.25 \pm j\ 12.7$ | .174 | .2963 | 12.702 | .019 |
| $-.25 \pm j\ 10.0$$ | .215 | .21664 | 10.003 | .024 |
| $-.05 \pm j\ 15.4$ | .151 | .3581 | 15.4 | .0006 |
| $-.15 \pm j\ 15.4$ | .152 | .3399 | 15.4 | .0097 |
| $-.35 \pm j\ 15.4$ | .152 | .305 | 15.4 | .0227 |
| $-.45 \pm j\ 15.4$ | .152 | .2877 | 15.403 | .0292 |
| $-.2 \pm j\ 12.32$* | .177 | .2891 | 12.321 | .016 |
| $-.3 \pm j\ 18.48$* | .128 | .2585 | 18.482 | .016 |

! The original system

* The perturbed system created using these eigenvalues has the same damping ratio as the original system

$ Local optimal sample rate only; i.e., there is another peak

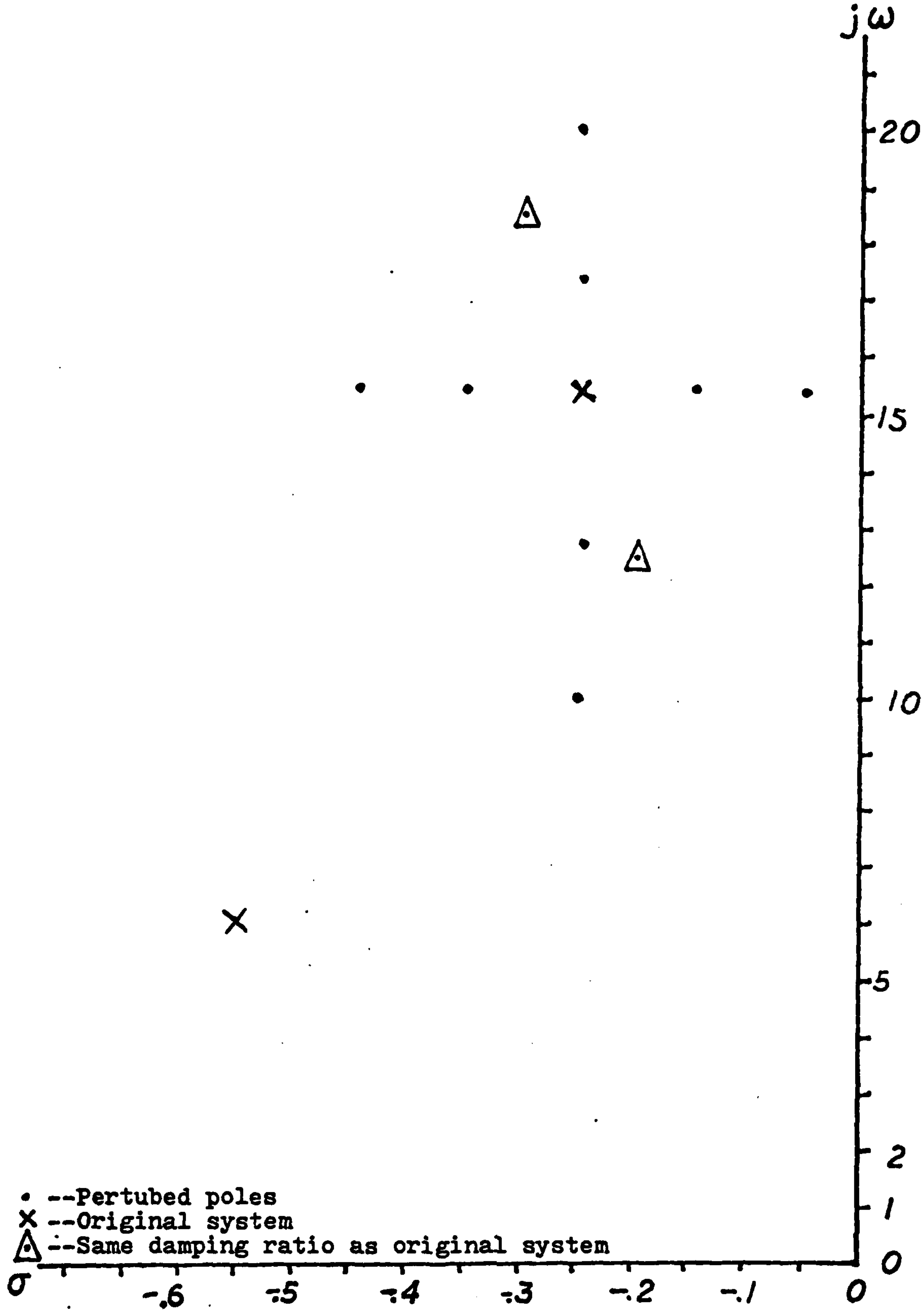
Looking at Figures 43 thru 52 one notices a trend appearing. When the real part of the eigenvalue is perturbed and the imaginary part is




ONE OVER CONDITION NUMBER VS SAMPLE RATE

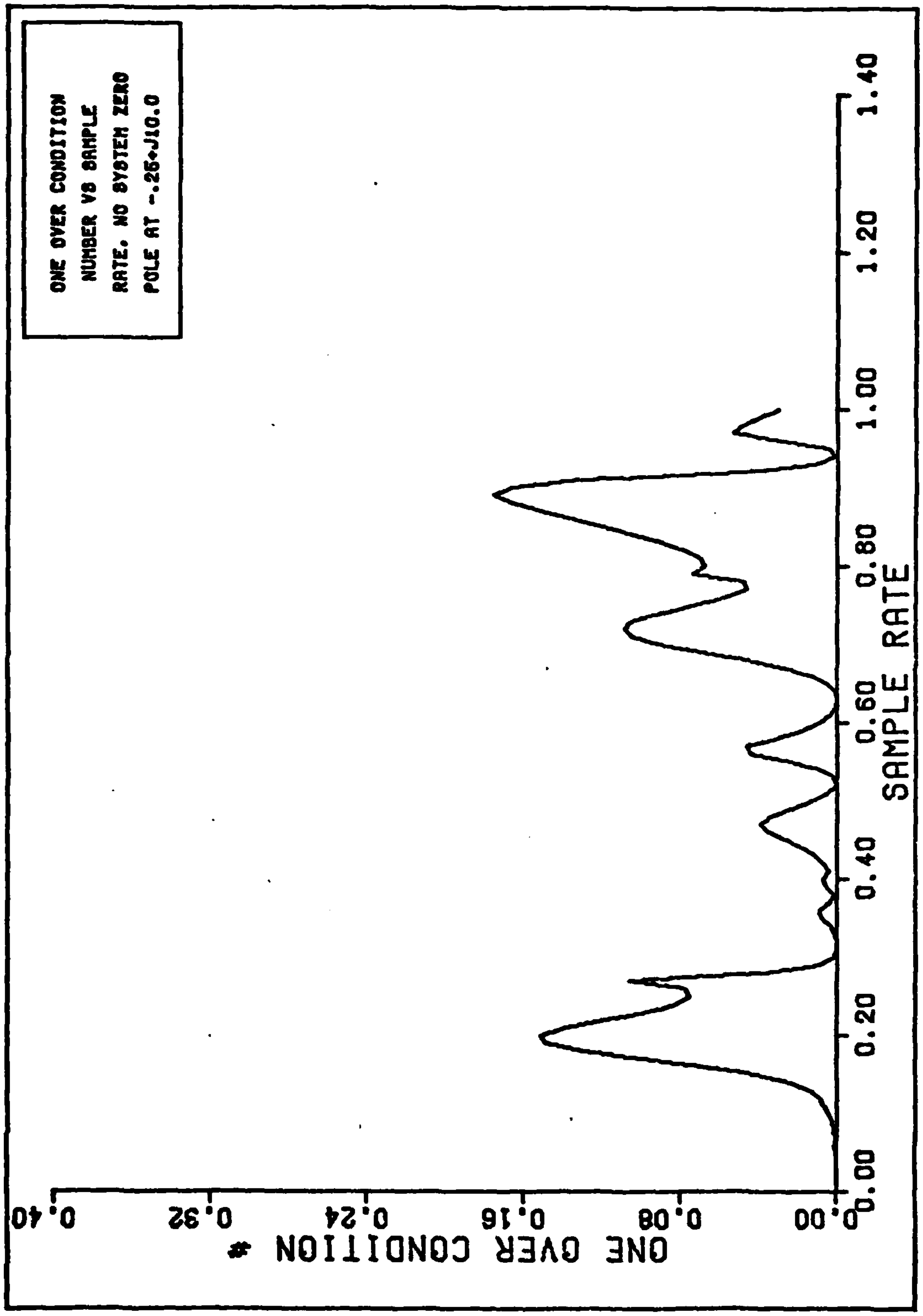


Fig. 41




Perturbed Eigenvalues for
Sensitivity Analysis

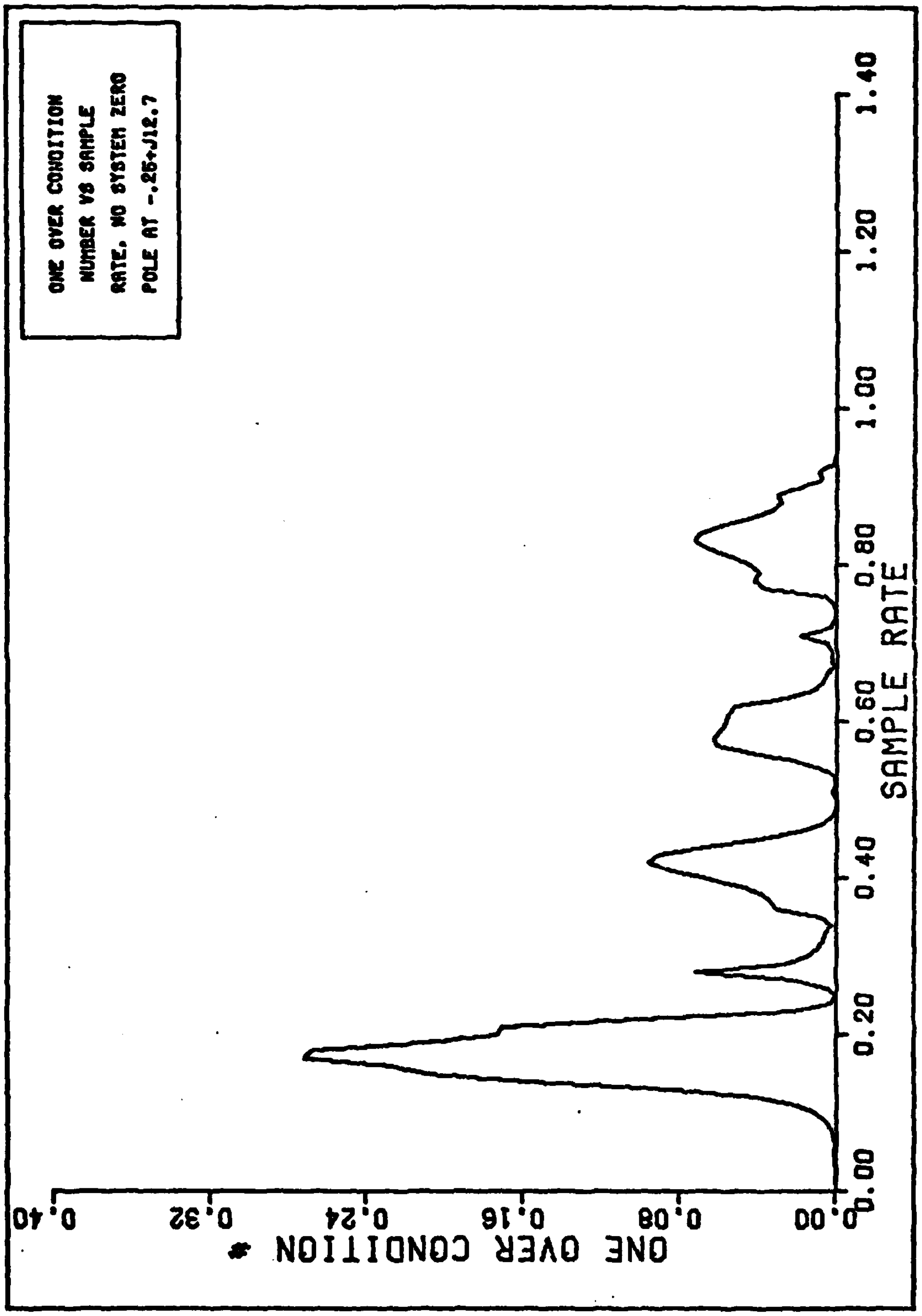
**Fig. 42**




ONE OVER CONDITION NUMBER VS SAMPLE RATE
WITH DOMINANT POLES AT $-.25\pm j10$.



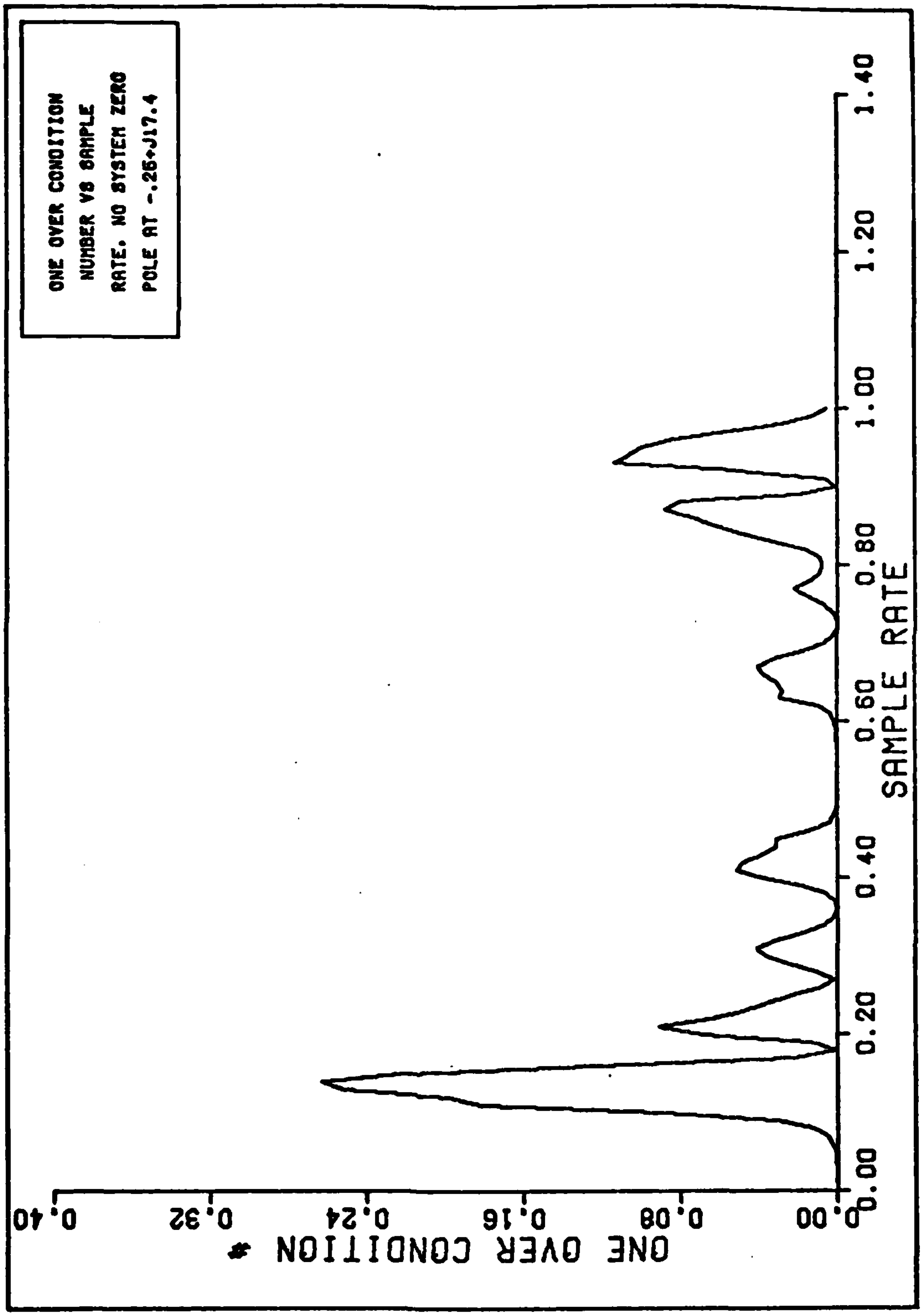
Fig. 43




ONE OVER CONDITION NUMBER VS SAMPLE RATE
WITH DOMINANT POLES AT $-.25 \pm j12.7$

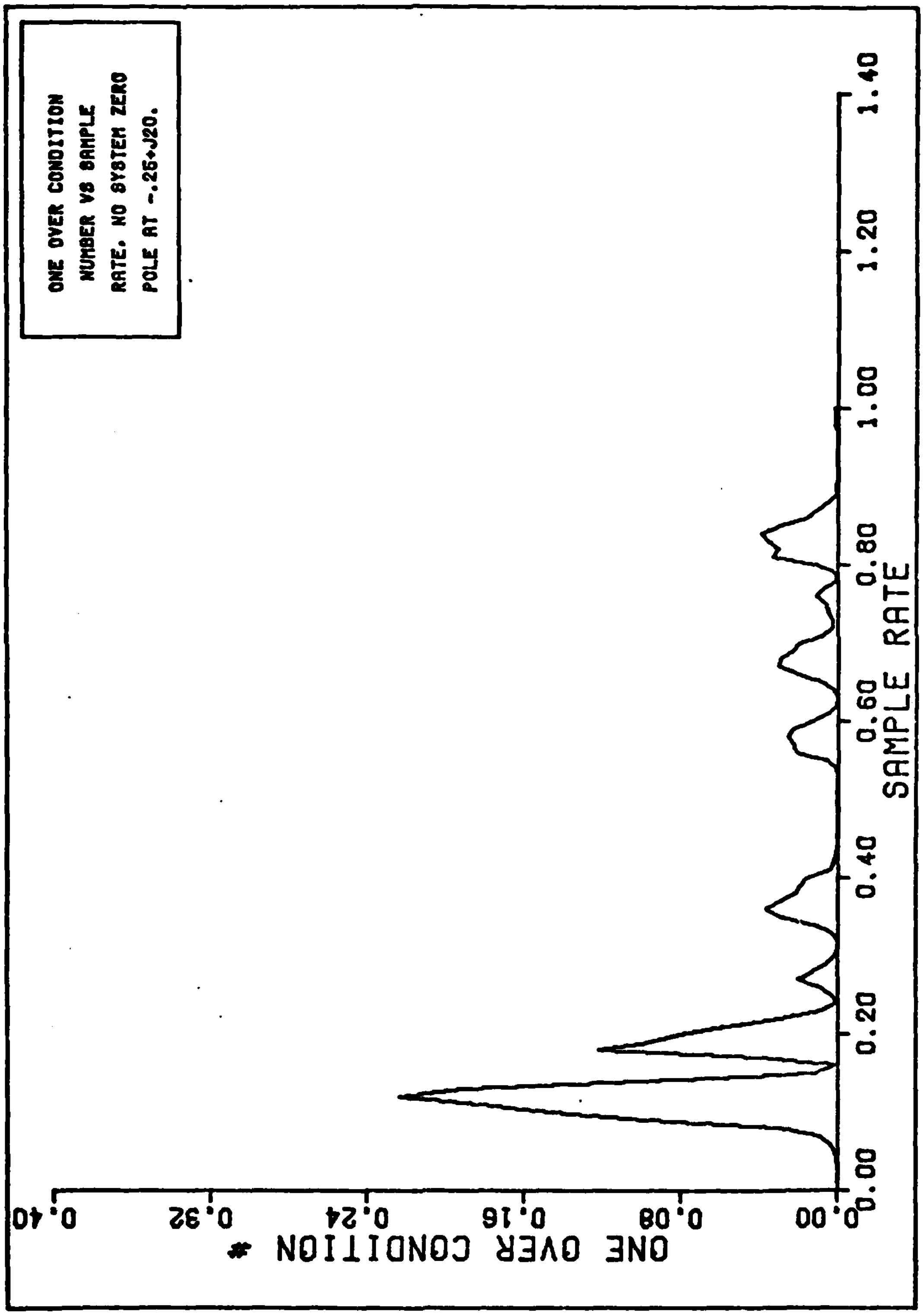


Fig. 44




ONE OVER CONDITION NUMBER VS SAMPLE RATE
WITH DOMINANT POLES AT -.25 ± j17.4

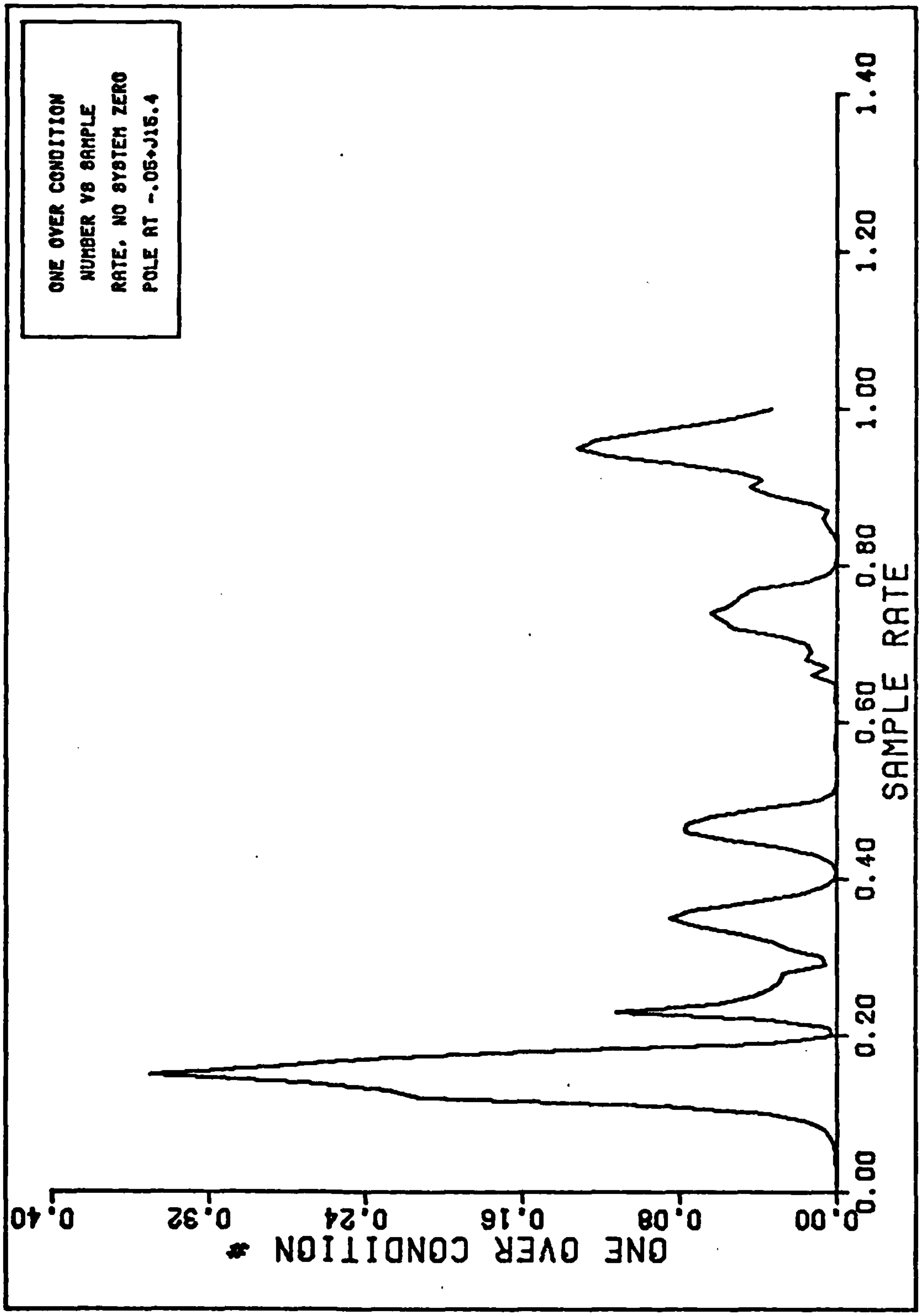
Fig. 45




ONE OVER CONDITION NUMBER VS SAMPLE RATE
WITH DOMINANT POLES AT $-.25 \pm j20$



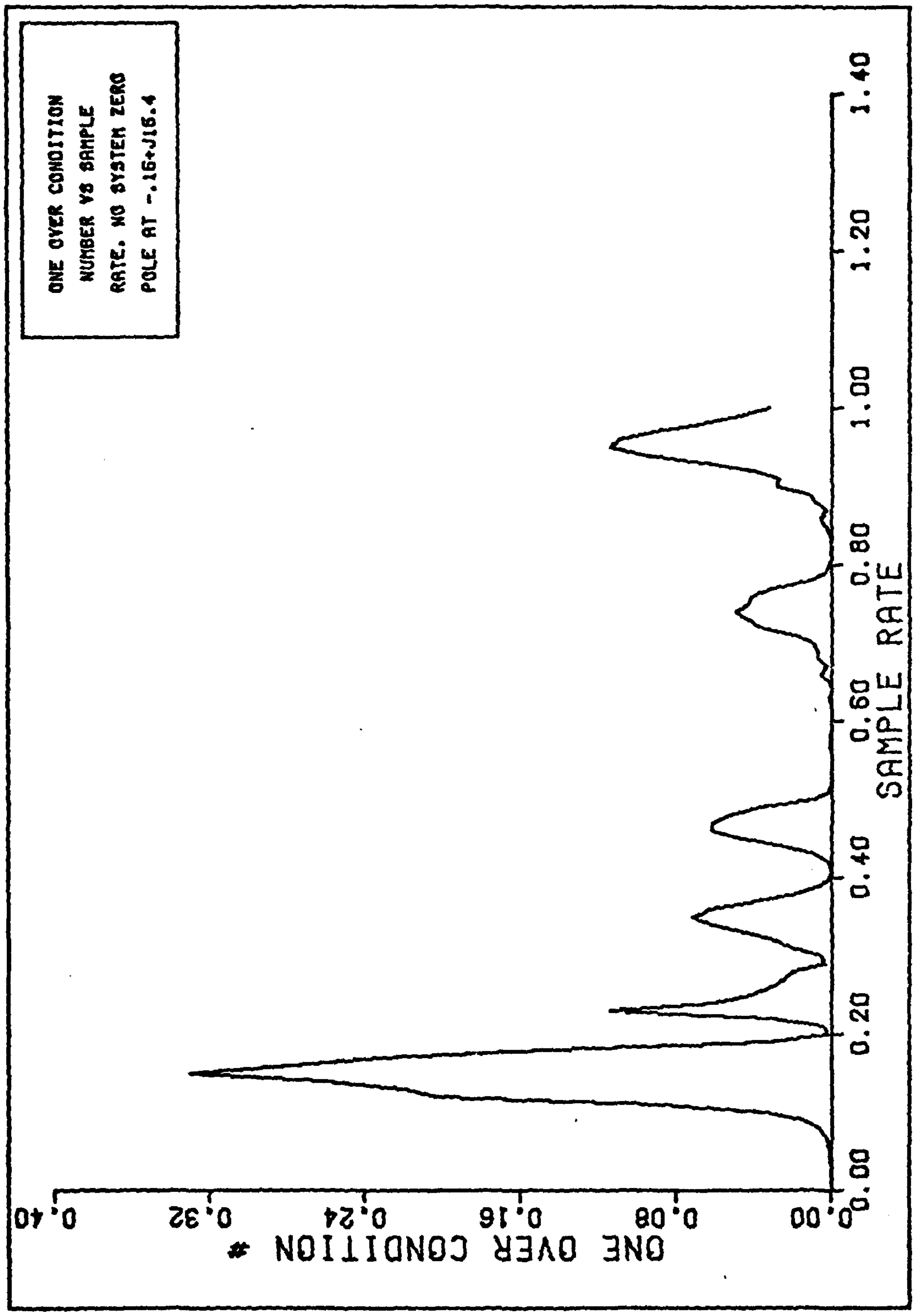
Fig. 46




ONE OVER CONDITION NUMBER VS SAMPLE RATE
WITH DOMINANT POLES AT $-.05 \pm j15.4$



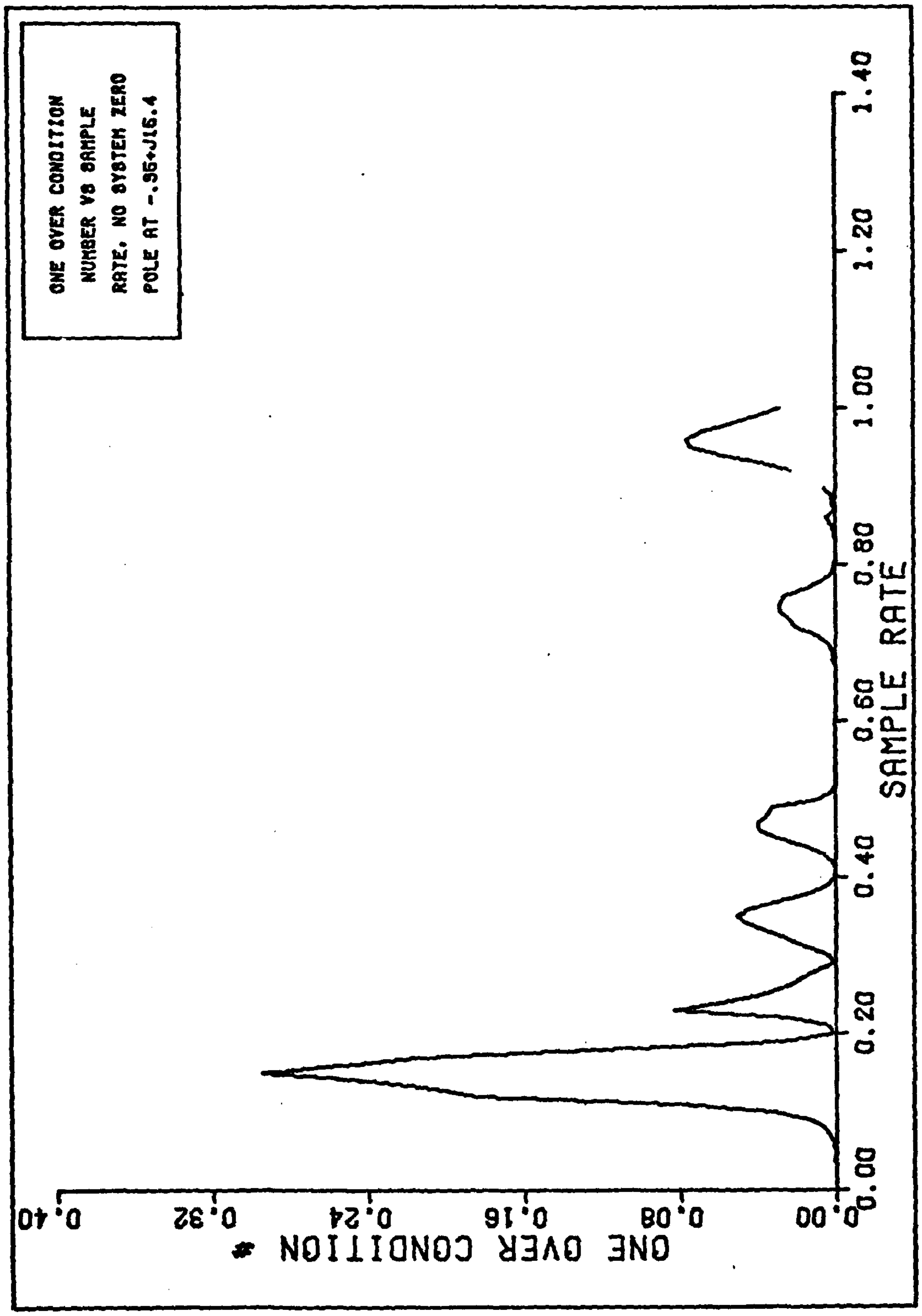
Fig. 47




ONE OVER CONDITION NUMBER VS SAMPLE RATE
WITH DOMINANT POLES AT -.15 ± j15.4



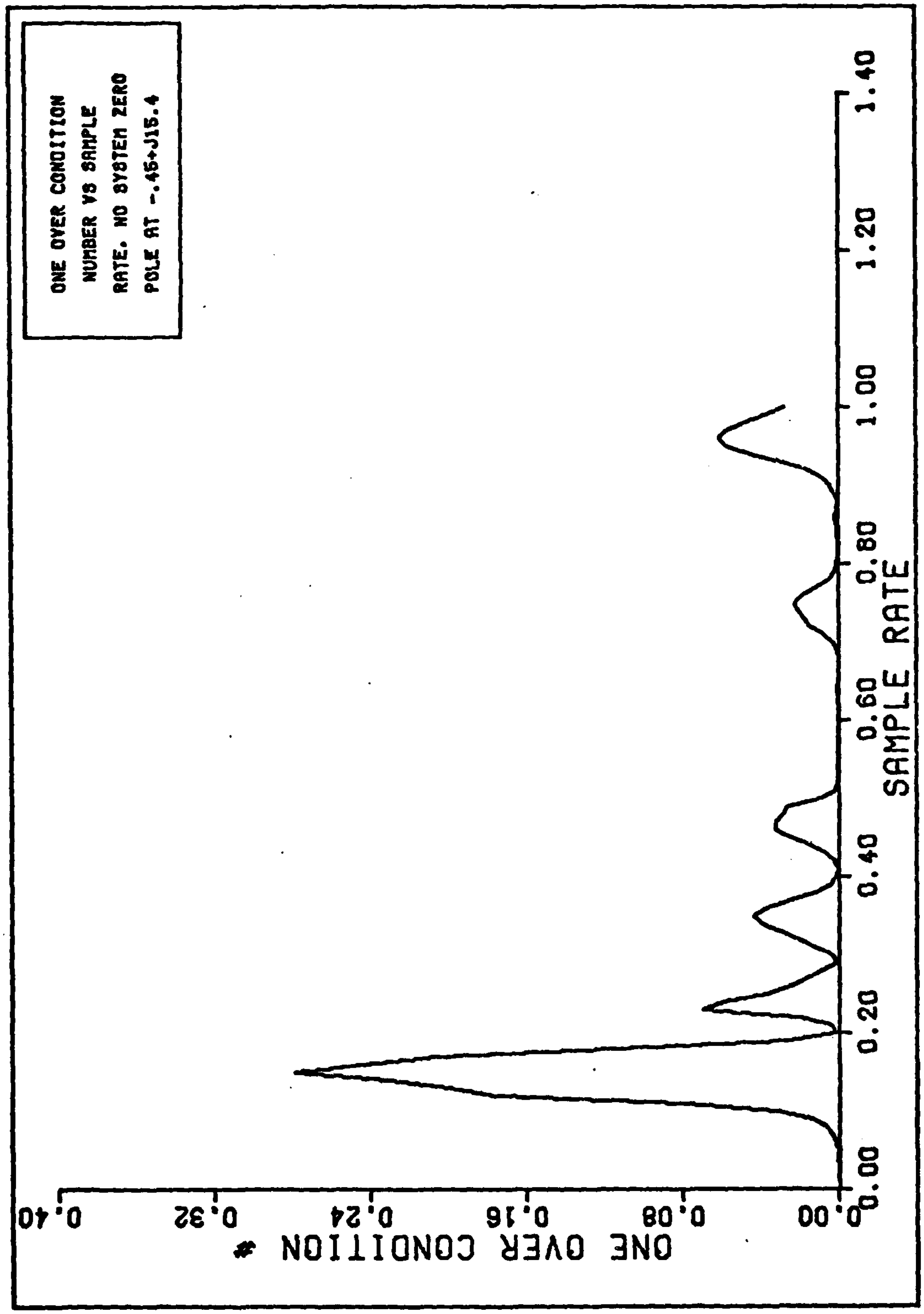
Fig. 48




**ONE OVER CONDITION NUMBER VS SAMPLE RATE**
WITH DOMINANT POLES AT $-.35 \pm j15.4$



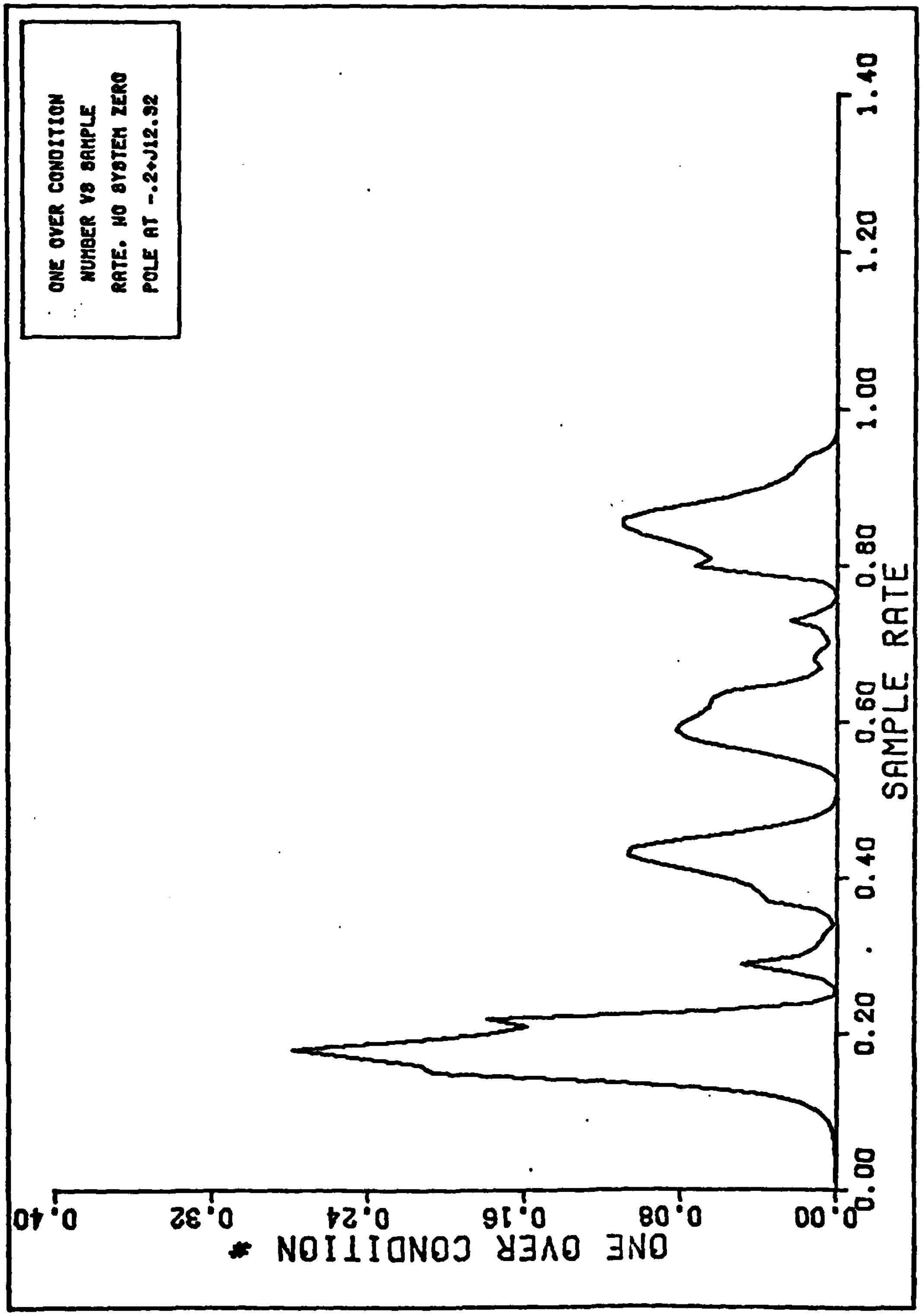
Fig. 49




ONE OVER CONDITION NUMBER VS SAMPLE RATE
WITH DOMINANT POLES AT $-.45 \pm j15.4$

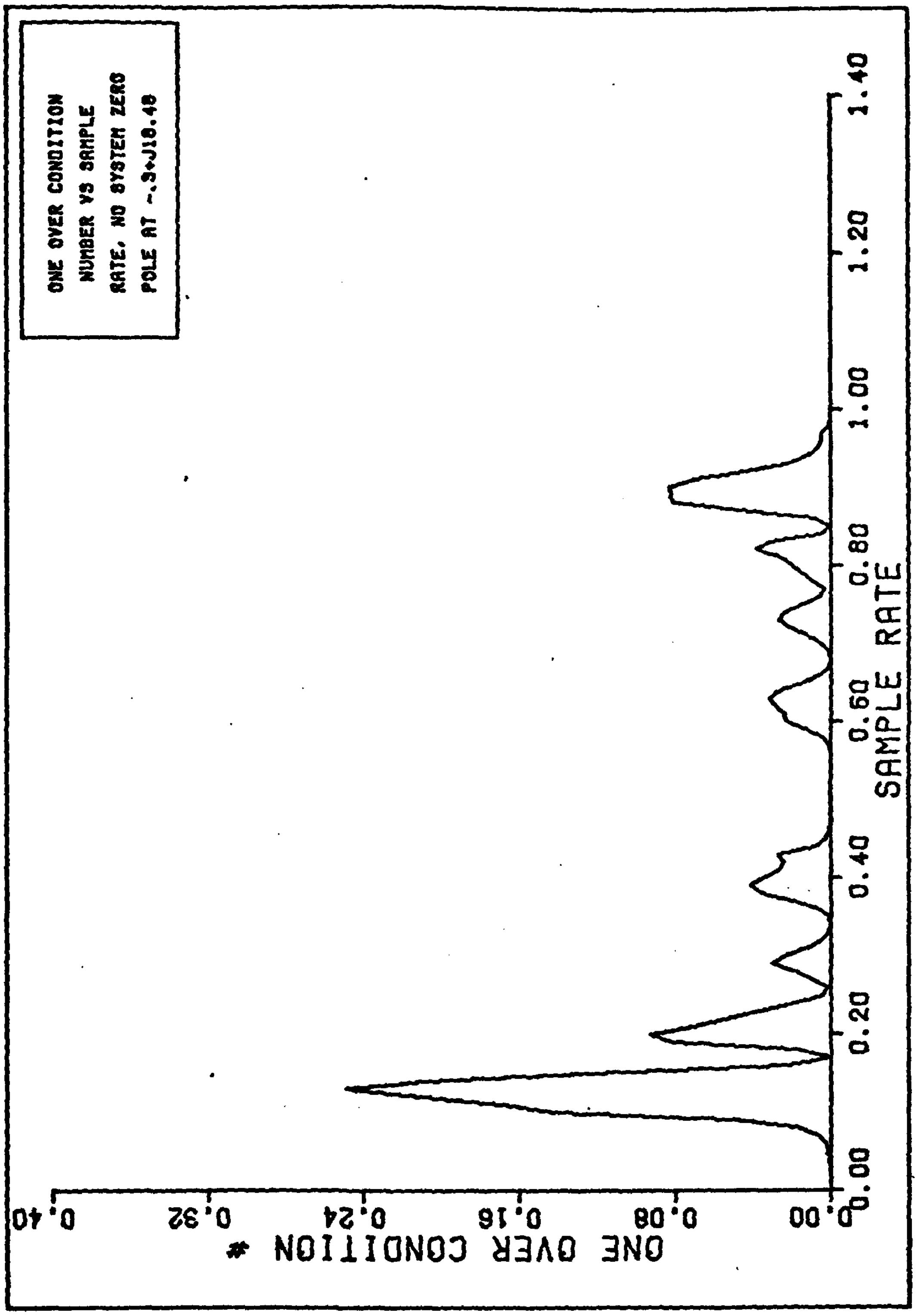


Fig. 50




ONE OVER CONDITION NUMBER VS SAMPLE RATE
WITH DOMINANT POLES AT -.2 $\pm$ j12.32

Fig. 51




ONE OVER CONDITION NUMBER VS SAMPLE RATE
WITH DOMINANT POLES AT $-.3 \pm j18.48$



Fig. 52

not, the effects upon the optimal sample rate is minimal. Conversely, when the real part is not perturbed and the imaginary part is perturbed, the shift of the optimal sample rate is considerable. This seems reasonable because the frequency of a system is very dependent upon the imaginary portion of the eigenvalues in a lightly damped system. This analysis shows the Hankel matrix condition number being sensitive to the frequency, Wn, of a system, but insensitive to the damping ratio of a system.

The next area of concern is the effect a system zero would have on the Hankel matrix. A system zero is added to the basic fourth order system giving us a new system, shown below.

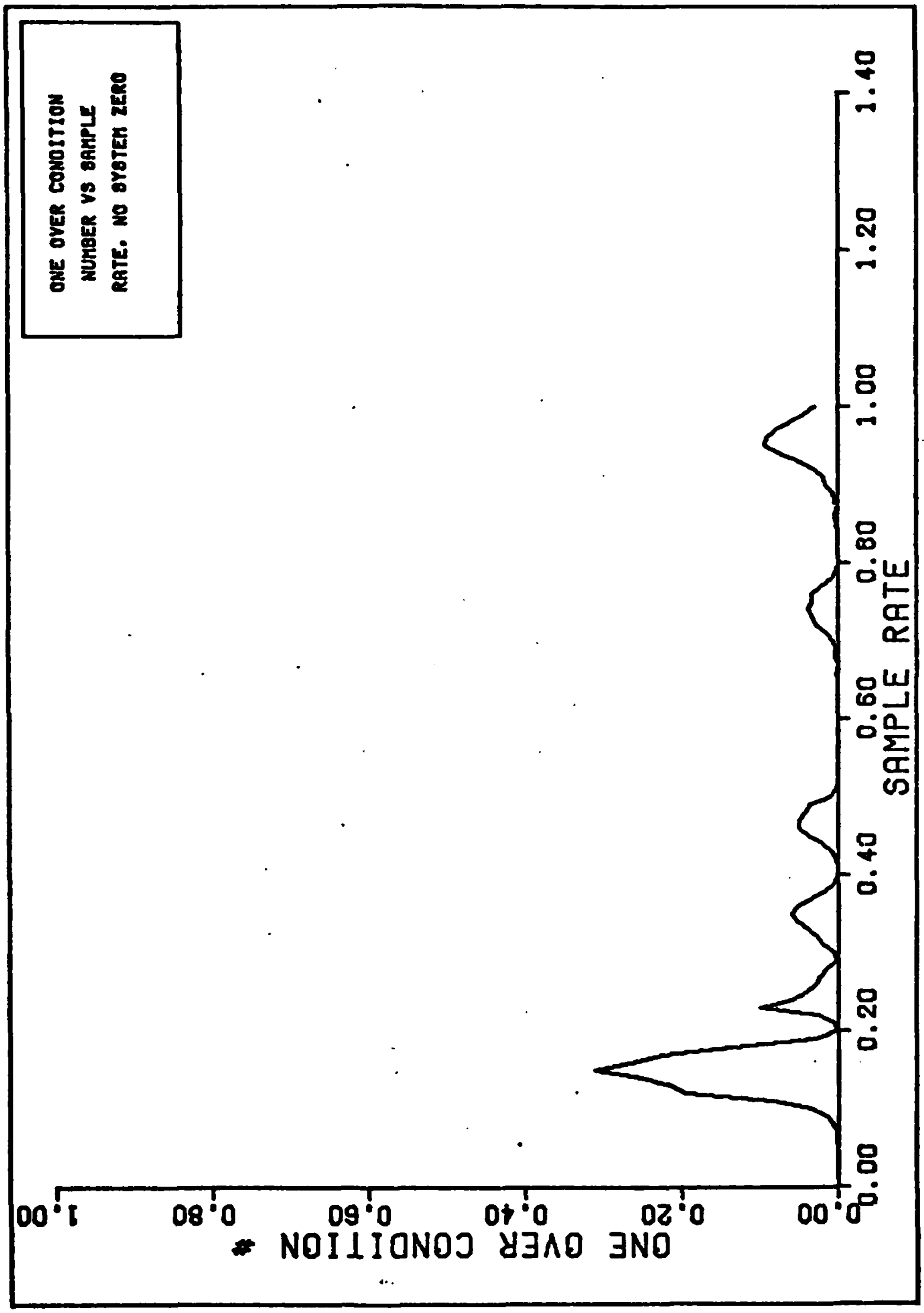
$$\frac{(S+\alpha)8611.7698}{S^4+1.6S^3+274.075S^2+279.096+8611.7698}$$

Alpha is varied from one to minus one to see the sensitivities in the Hankel matrix. Table 11 shows the optimum sample rate and the corresponding reciprocal condition value for various values of alpha.

TABLE 11

| ALPHA | "OPTIMAL" SAMPLE RATE | RECIPROCAL CONDITION NUMBER |
|---|---|---|
| 1.0 | .147 | .6234 |
| .3 | .156 | .6336 |
| .2 | .156 | .6352 |
| 0.0 | .156 | .6395 |
| -.2 | .156 | .6438 |
| -.3 | .156 | .6456 |
| -1.0 | .153 | .6526 |

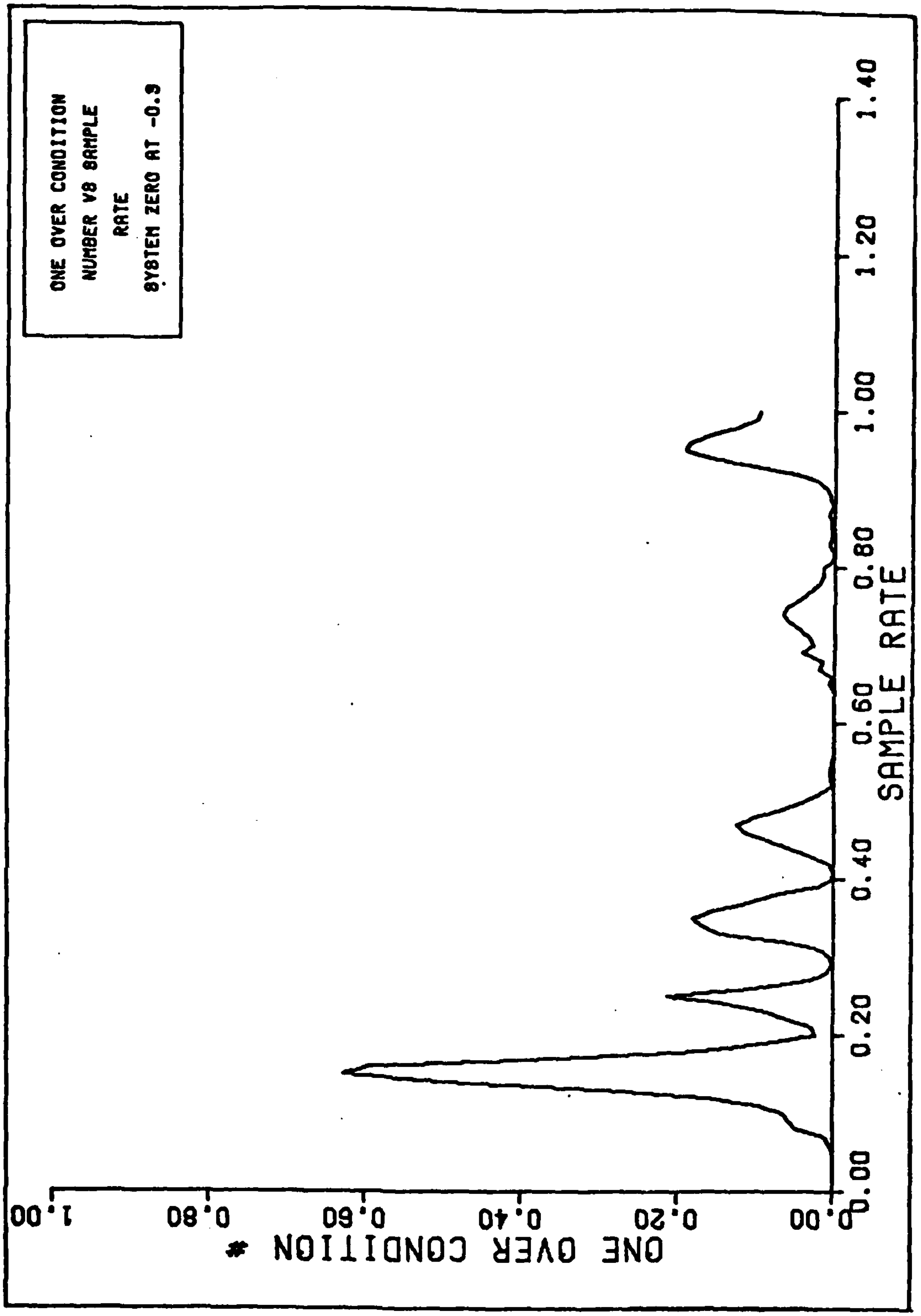
The original system condition number plot is again in Figure 53. For alpha's of .3, 0.0, and -.3, plots of the reciprocal condition number versus sample rate are shown in Figures 54, 55 and 56, respectively.




ONE OVER CONDITION NUMBER VS SAMPLE RATE



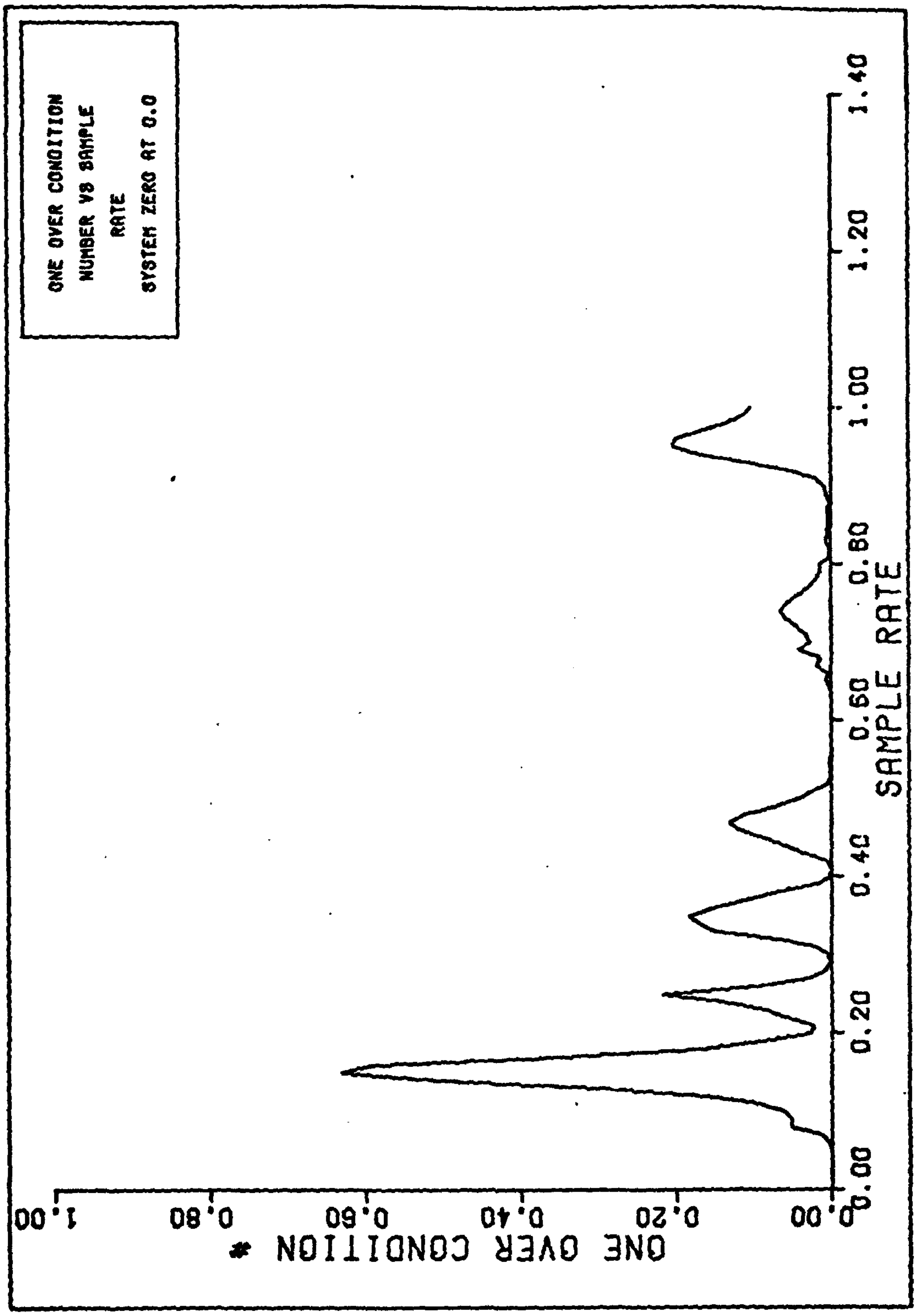
Fig. 53




ONE OVER CONDITION NUMBER VS SAMPLE RATE
WITH SYSTEM ZERO AT -.3



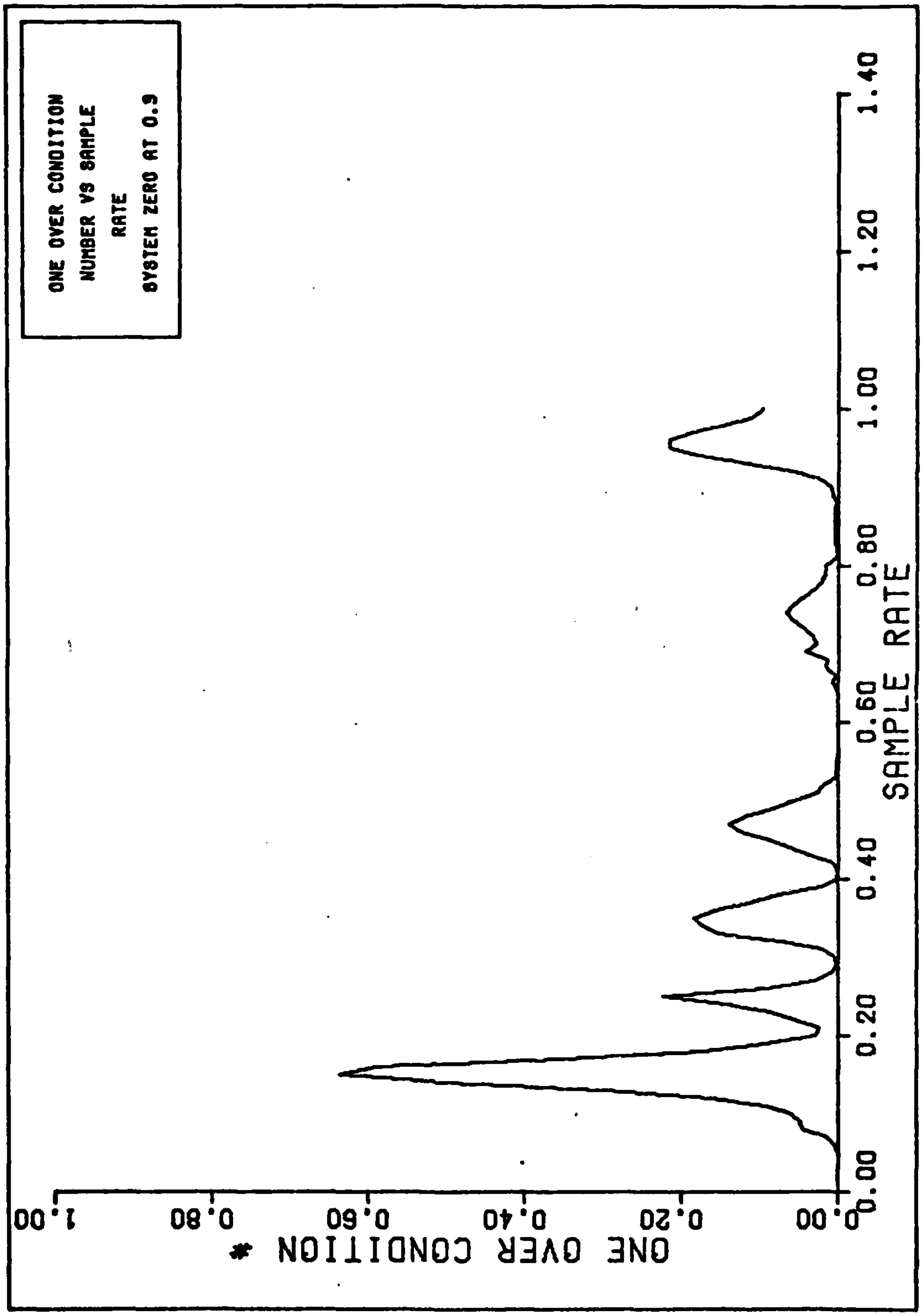
Fig. 54




ONE OVER CONDITION NUMBER VS SAMPLE RATE
WITH SYSTEM ZERO AT 0.0



Fig. 55




ONE OVER CONDITION NUMBER VS SAMPLE RATE
WITH SYSTEM ZERO AT +0.3



Fig. 56

The most obvious change is the value of the maximum reciprocal condition number. The value has about doubled from .3 for the no zero system to .6 for the system with a zero. Also observed is that variation on the optimal sample rate itself is very small and it changes very little as the zero shifts, even when the zero is in the right half plane. The system is then a nonminimum phase system, but there seems to be little influence upon the characteristics of the conditioning of the Hankel matrix. If anything, the maximum reciprocal condition number is increasing in value as we proceed further into the right half plane! If the condition numbers value is any indication of robustness, this indicates that a system with a zero will be more robust than the same system without a zero. From this analysis, using a lightly damped system, the Hankel matrix is sensitive to the pole locations of the system, but it is relatively insensitive to the location of the system zeros.

Minimum Phase versus Nonminimum Phase

This section examines whether or not a nonminimum phase system will present any problems when being controlled by OPDEC. Two systems, one being nonminimum phase, the other minimum phase, are implemented under conditions of model errors and input and/or state noise. The two systems are:

System A (Minimum Phase)

$$\frac{8611.7698(S+.3)}{S^4+1.6S^3+274.075S^2+279.096S+8611.7698}$$

System B (Nonminimum Phase)

$$\frac{8611.7698(S-.3)}{S^4+1.6S^3+274.075S^2+279.096S+8611.7698}$$

From Table 11, both System A and B have an optimal sample rate of .156 seconds. Their sampled outputs and control inputs at the optimal sample is shown in Figures 57 and 58. Surprisingly, the sampled outputs and control inputs are exactly the same for a minimum phase system and a nonminimum phase system. This is caused by the zeros in both systems having the same magnitude. Input noise, then state noise, and finally input and state noise are added to the system. The sampled output with input noise added at a sample rate of .156 can be seen in Figure 59. Again the output is exactly the same for a nonminimum phase system as for a minimum phase system. This fact is present when state noise is added and also when both types of noises are added. Other sample rates are tried and the outputs are identical for system A and B. Similar to previous results, a faster than optimum sample rate gives better robustness characteristics when noise is disrupting the system (Figures 60 and 61).

The next area of concern is if system A and system B have the same properties when an error model is used for control. The two 10% perturbed models are:

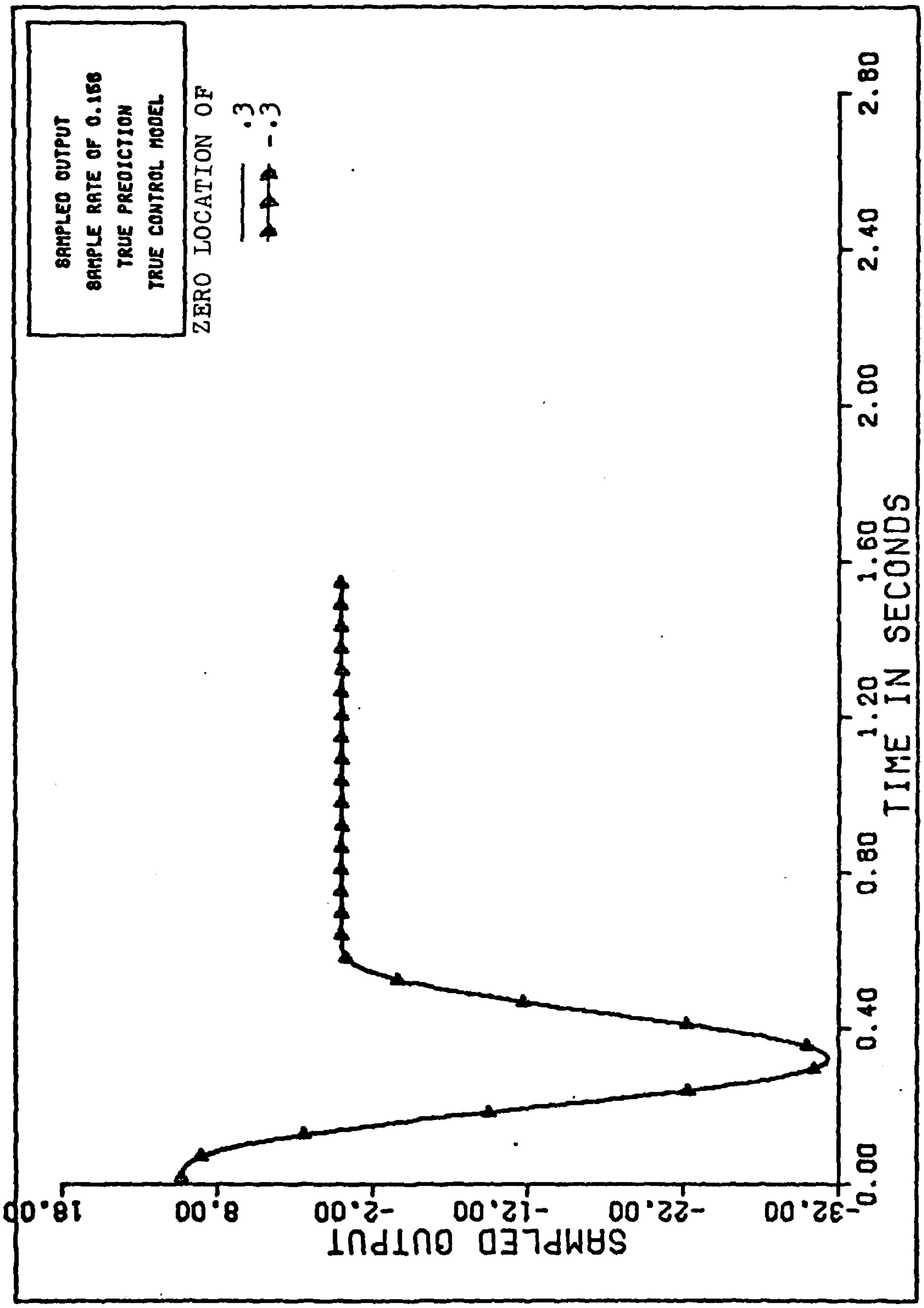
System A'

$$\frac{8474(S+.3)}{S^4+1.66S^3+317.1S^2+360.6S+8474}$$

System B'

$$\frac{8474(S-.3)}{S^4+1.66S^3+317.1S^2+360.6S+8474}$$

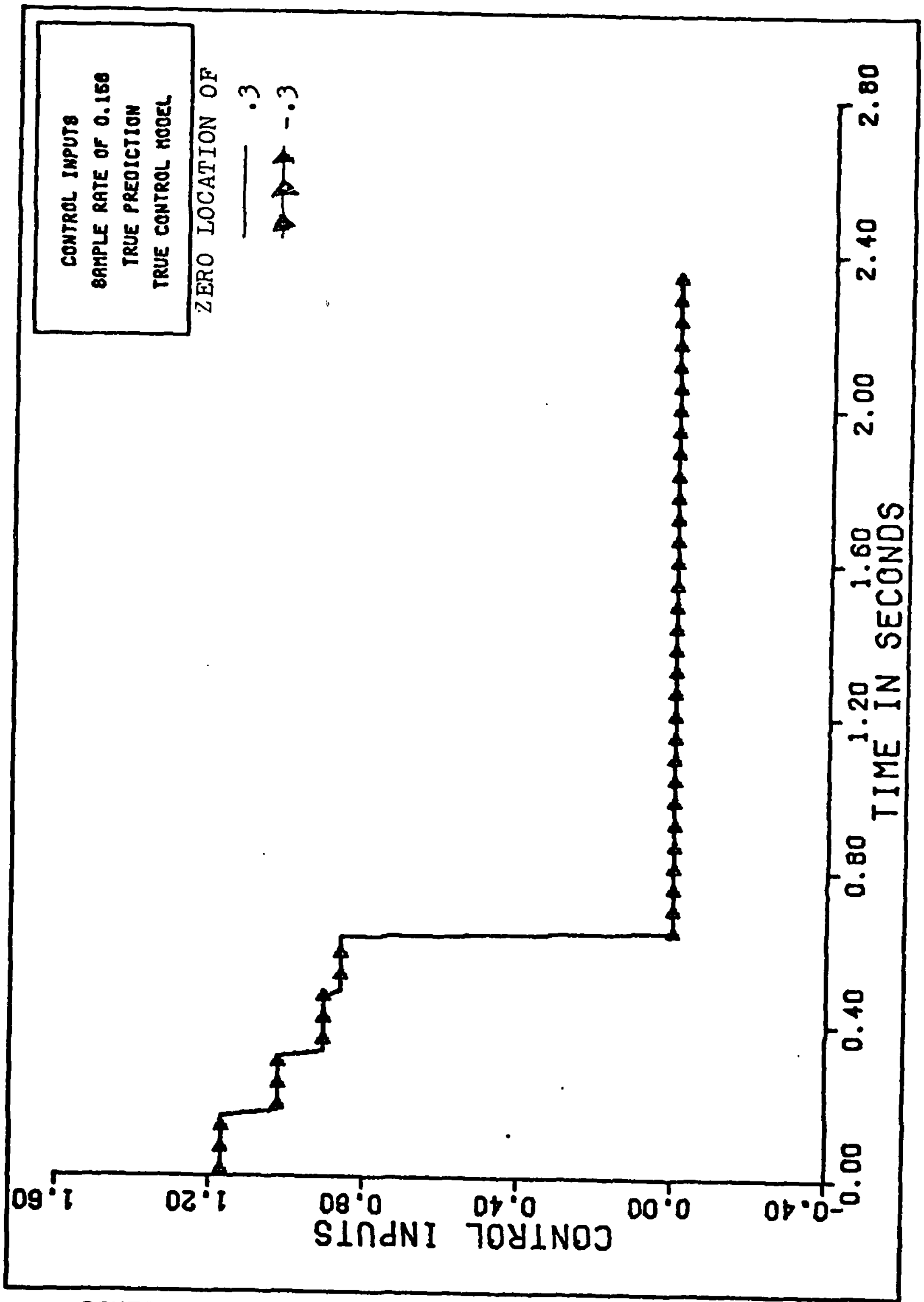
The denominator of the perturbed systems is the same as the previous 10% perturbed models denominator used earlier in this report. The output response and control inputs when using the perturbed models to control




SAMPLED OUTPUT WITH NO NOISE ADDED AT A
SAMPLE RATE OF .156 SECONDS



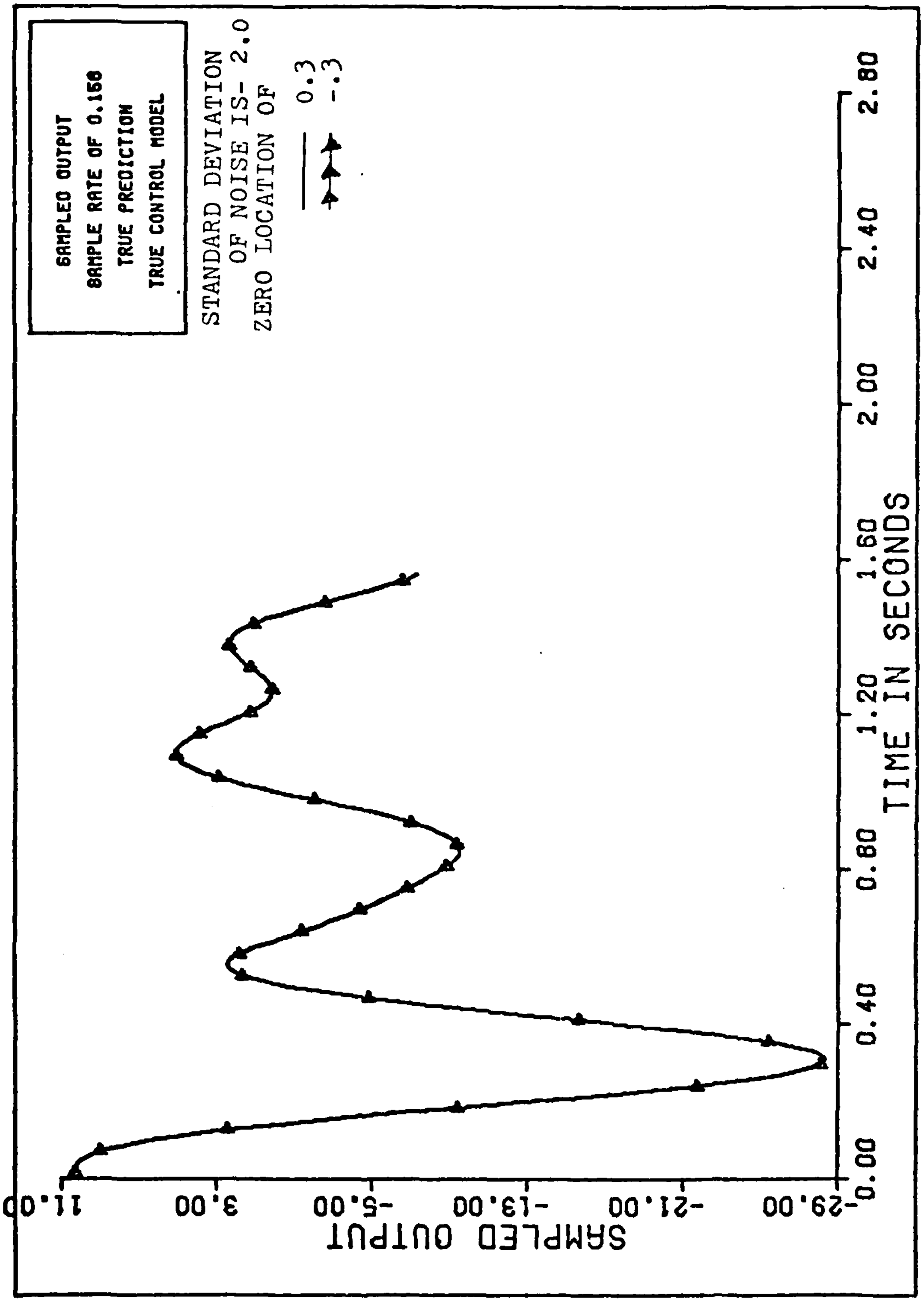
Fig. 57




CONTROL INPUTS WITH NO NOISE ADDED AT A SAMPLE RATE OF .156 SECONDS



Fig. 58




SAMPLED OUTPUT WITH STATE NOISE ADDED AT A SAMPLE RATE OF .113 SECONDS.

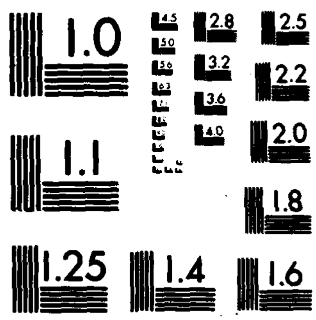


Fig. 59

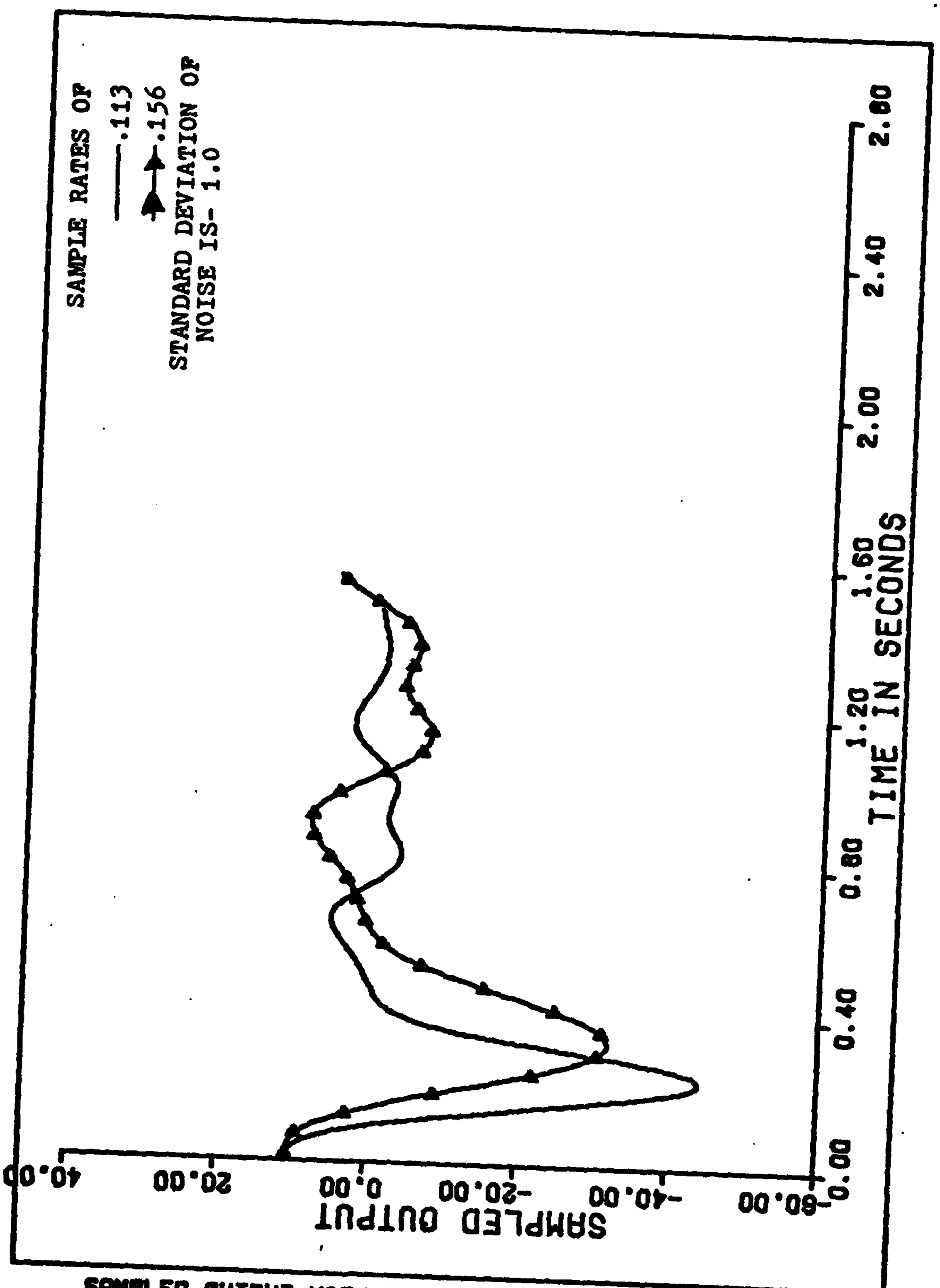
AD-A085 709 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 12/1
ROBUSTNESS STUDIES OF OUTPUT PREDICTIVE DEAD-BEAT CONTROL FOR W--ETC(U)
DEC 79 E H KIRKWOOD
UNCLASSIFIED AFIT/GA/EE/80-1 NL

2 of 2

END
DATE FILMED
7-80
DTIC



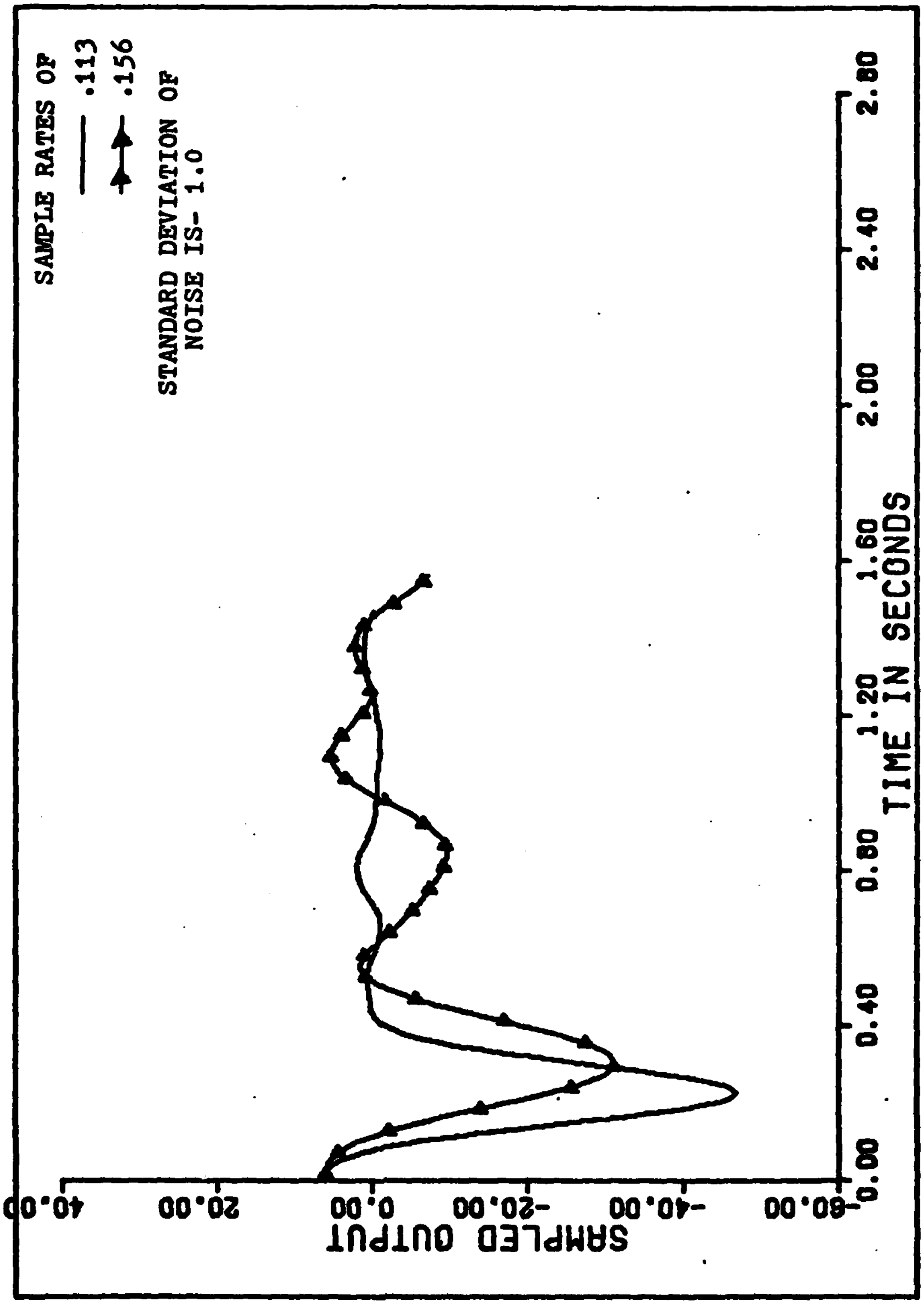

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A




SAMPLED OUTPUT WITH INPUT NOISE ADDED AT
SAMPLE RATES OF .113 and .156 SECONDS.



Fig. 60




SAMPLED OUTPUT WITH STATE NOISE ADDED AT
SAMPLE RATES OF .113 and .156 SECONDS



Fig. 61

the true system are identical. Figure 62 shows the sampled outputs at sample rates of .113 and .152 using this perturbed model.

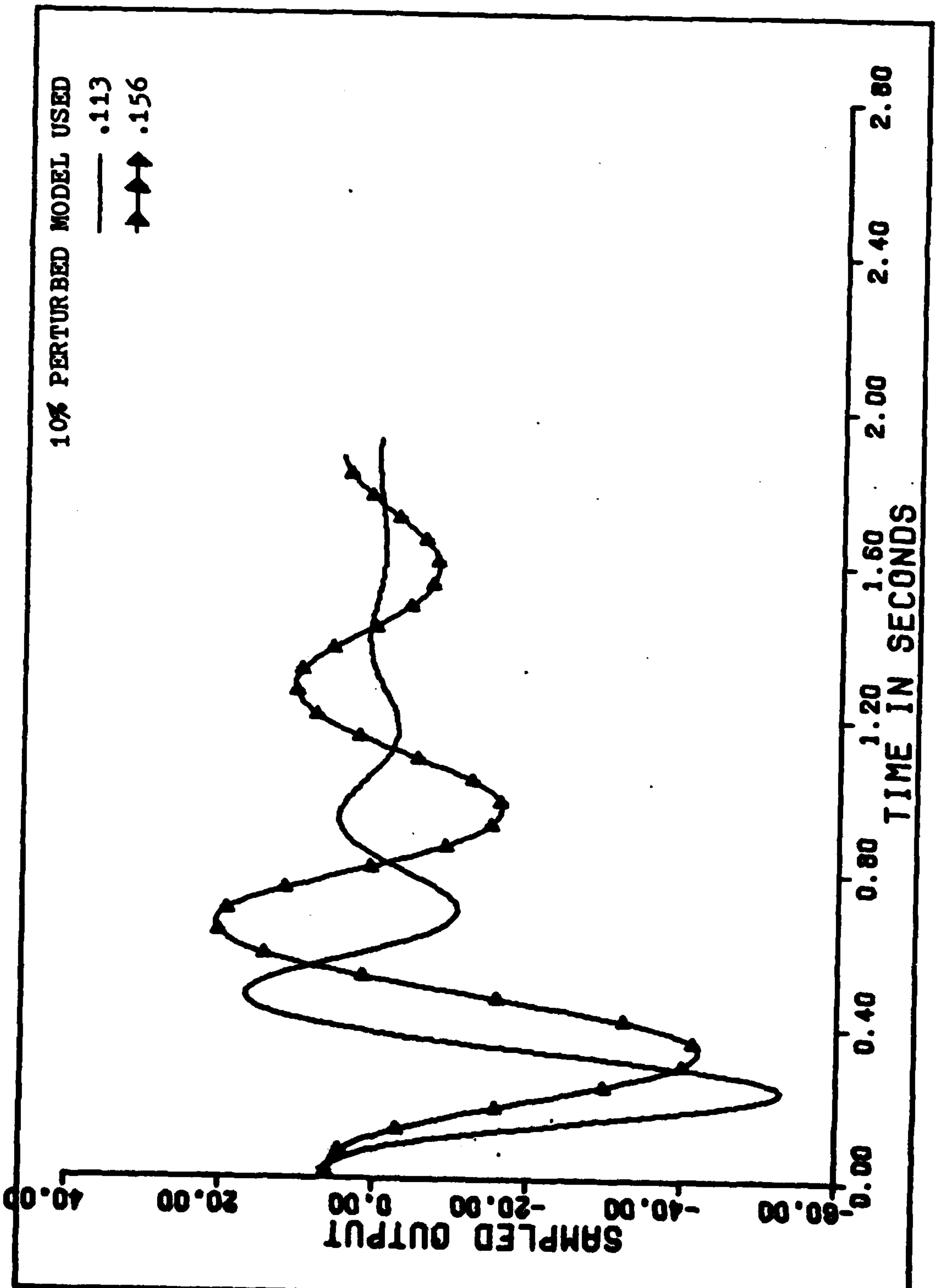
From this analysis, OPDEC has no problems controlling a nonminimum phase system. Also, the robustness of OPDEC is not degraded when this type system is being controlled. OPDEC is fairly insensitive to zero placement.

This insensitivity to the system zero location was interesting enough to try and control system B using system A and A' as the perturbed model. When system A was used, the output response looked exactly like the original dead-beat response (Figure 57). When system A' was used, the output response looked exactly like Figure 62. The next step was to try and use the original fourth order SISO system with no zeros to control system B. The systems response was unstable. What this analysis shows is that the zero location is not a factor in using OPDEC as a regulator.

## Discussion of 4th Order Results

From this fourth order analysis, we see that sample rates faster than the optimum have better output characteristics when noise is corrupting the system. Also, some of the faster sample rates seem to have better robustness characteristics overall. The condition number is a helpful guide in selecting general areas of sample rates where the system is more robust. Using the condition number to find a specific "optimal" sample rate, without doing the closed loop robustness simulation studies, is definitely the wrong approach. Robustness and closed loop system performance is just too complex of an issue to hope that there would be a single panacea for all aspects of the problem.

Areas of high sensitivity of OPDEC seem to be the poles of the system being controlled, but OPDEC appears to be insensitive to systems




SAMPLED OUTPUT WITH NO NOISE ADDED USING THE 10% PERTURBED MODEL AT SAMPLE RATES OF .113 and .156 SECONDS.



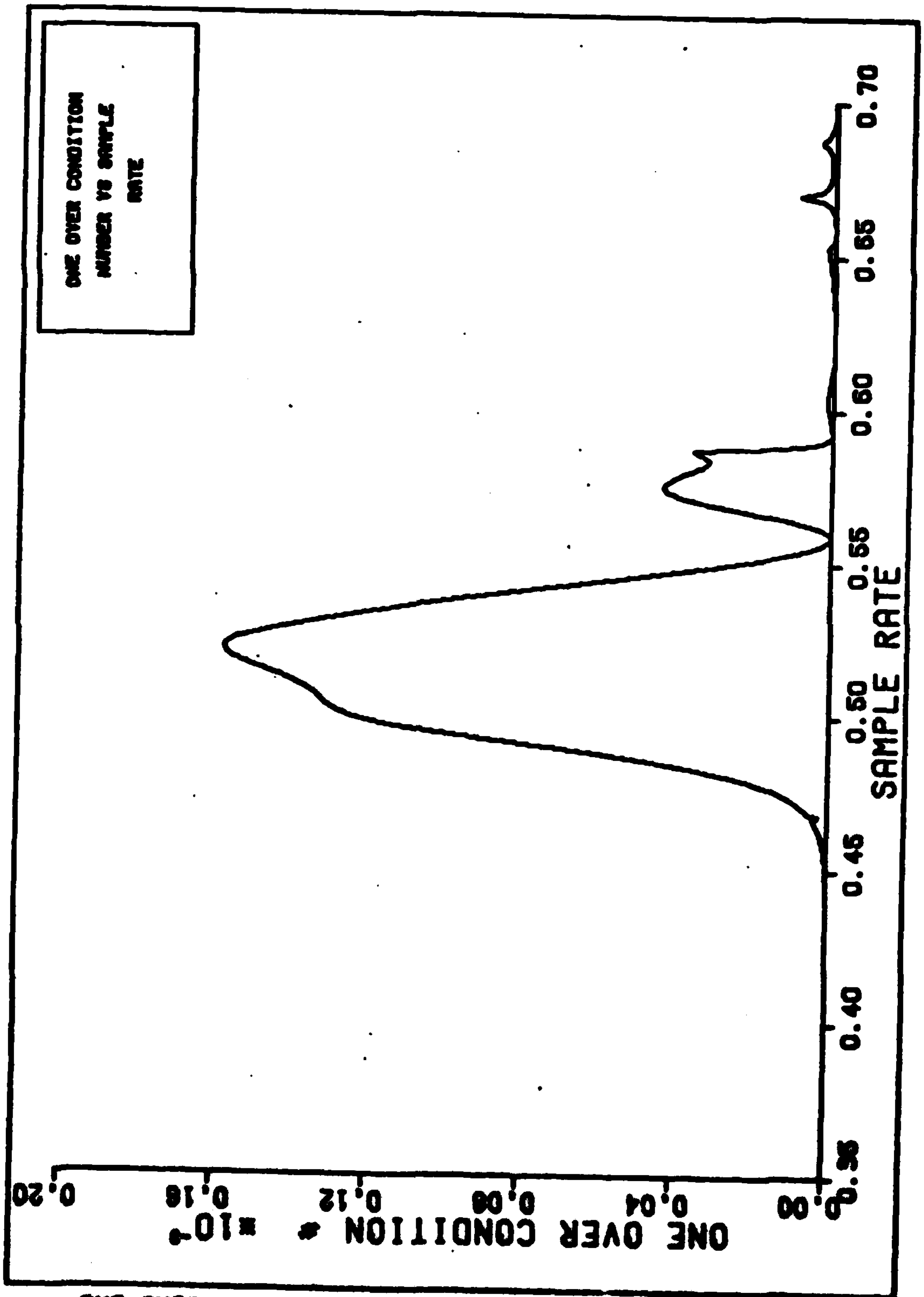
Fig.62

zero location. As far as the location of the poles are concerned, OPDEC is more sensitive to the "frequency" of the pole as compared to the damping ratio.

Procedure and Results with 10th Order SISO System

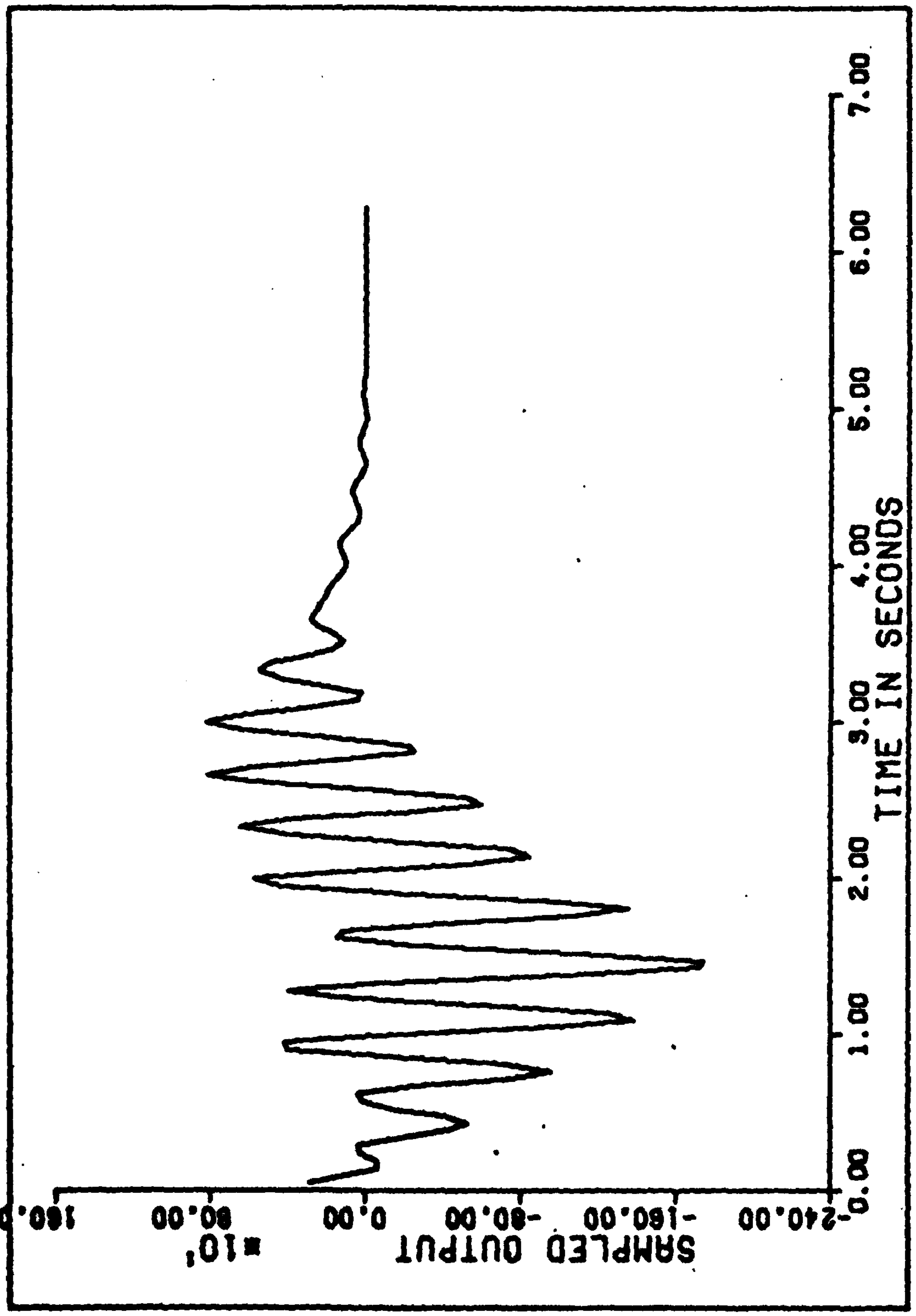
In this section, OPDEC is implemented using the 10th order system discussed earlier. Figure 63 shows the reciprocal condition number versus sample rate. The "optimum" sample rate is .523 and the reciprocal condition number is 0.000158. The sampled output at a rate of .523 is shown in Figure 64. The output is very oscillitory but, as the theory states, in ten steps of the dead-beat control the output and the states are brought identically to zero.

What was tried at this point was to control the 10th order system with a smaller ordered system. The full ordered Hankel matrix was created (10 X 10). Then an option was exercised in the program that lets the user reduce the size of the Hankel matrix just before it is inverted for use in closed loop control. Then the reduced order Hankel matrix is inverted and used to determine a control input using OPDEC's control law. When trying to control the true 10th order system with a 9th order system the output for sample rate of .523 is shown in Figure 65. For the same sample rate the order of the Hankel matrix was reduced to an 8th order system. The sampled output can be seen in Figure 66. Some other sample rates were chosen and the same procedure was used. The outputs for all the other selected sample rates went unstable, even for the 9th order system controlling the full 10th order system. The condition number is helpful in determining a sample rate in which a reduced order model is being used to determine control inputs.



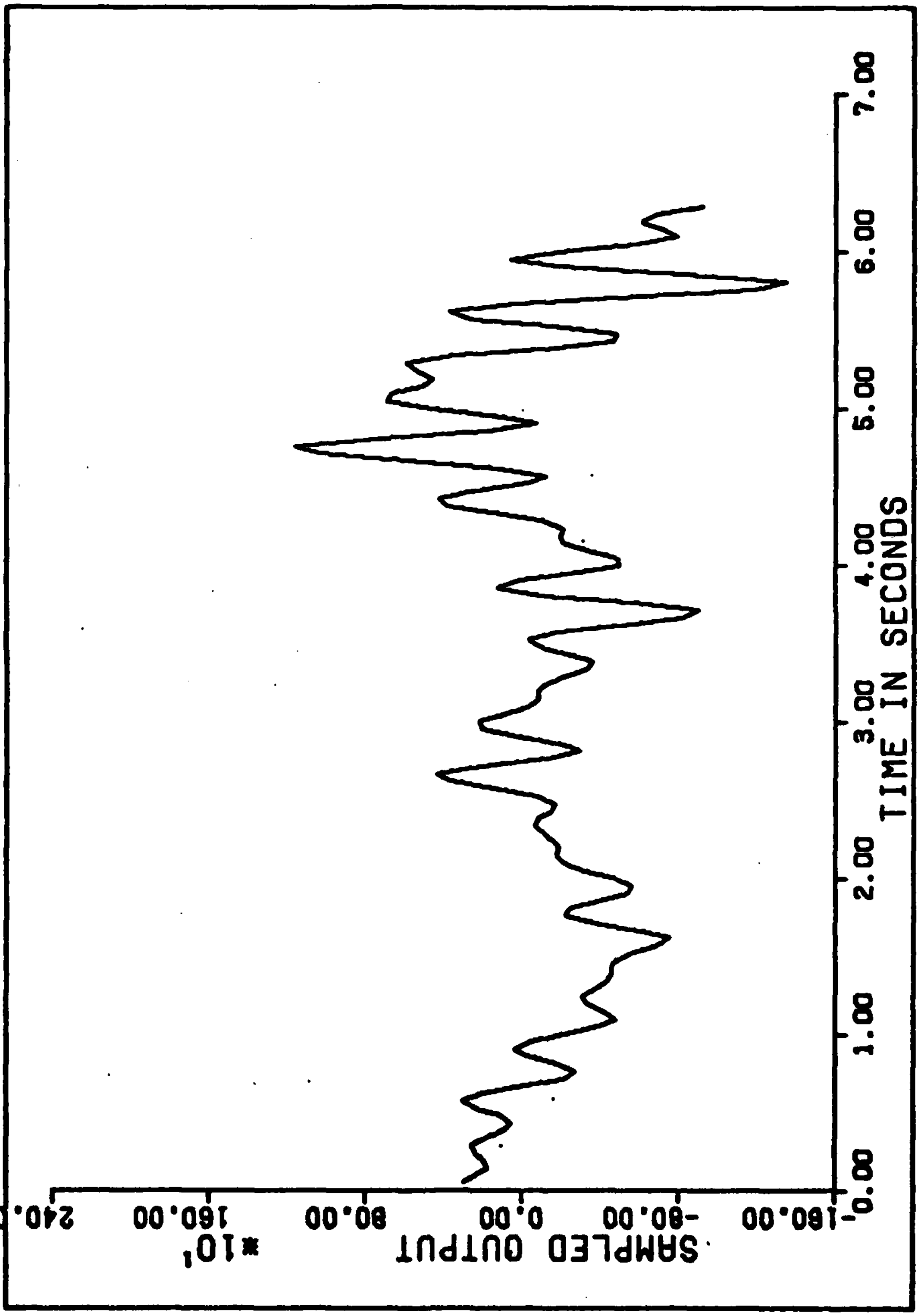

ONE OVER CONDITION NUMBER VS SAMPLE RATE

Fig. 63



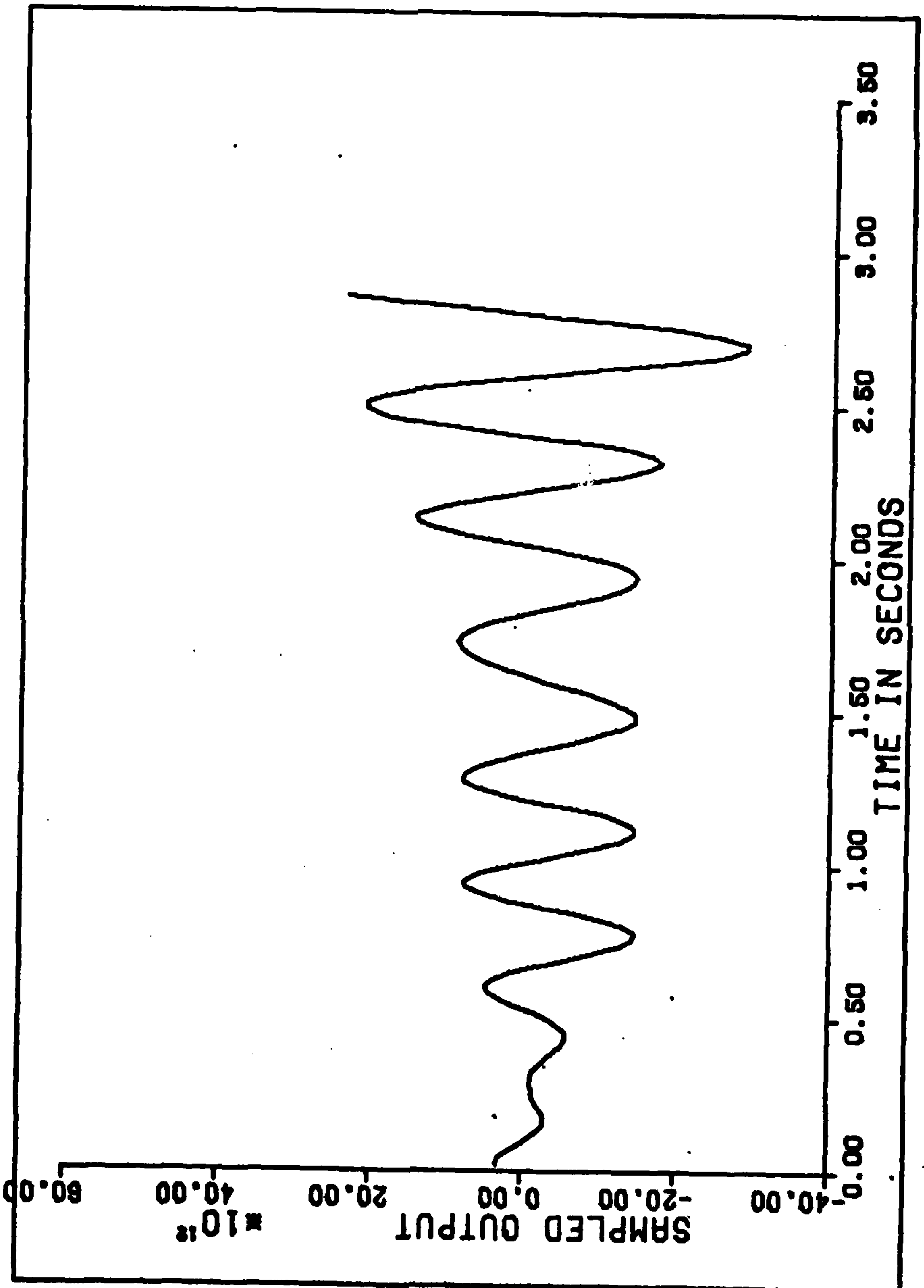

SAMPLED OUTPUT WITH NO NOISE ADDED AT A SAMPLE RATE OF .523 SECONDS.

Fig. 64




SAMPLED OUTPUT WITH NO NOISE ADDED With A 10th Order True System Being Controlled With A 9th Order system.

Fig. 65




SAMPLED OUTPUT USING AN 8th ORDER SYSTEM TO CONTROL A 10th ORDER SYSTEM.



Fig. 66

## IV Conclusions and Recommendations

From this analysis one can see that the sample rate can enhance the robustness properties of OPDEC. The sample rate which minimizes the condition number does yield controls with good magnitude properties. If anything, this fact causes the response to be more sensitive to input noise if the strength of the noise is held constant. Thus, the so called optimum sample rate is not necessarily the same sample rate that gives the best enhancement to OPDEC's robustness.

The group of sample rates which has good robustness properties when input noise was the only disruptive element is not the same group of sample rates that exhibit acceptable robustness when model mismatch exists. This means that the overall characteristics of OPDEC can be tailored for specific operating environments, by just selecting the proper sample rate. The problem then becomes one of selection of the proper sample rate to "tune" the algorithm for the particular set of noise environment and model errors which is expected to be encountered. There does not appear to be an analytic method for this proper sample rate. Rather, the robustness performance appears to be an issue which is best analyzed through simulation studies.

OPDEC's sensitivity to the frequency of the system and its operating sample rate could be caused from the selection of the system to control. It is recommended that future studies see if OPDEC has the same characteristics when implemented on other system, particularly on those which are not so lightly damped as the one studied here. Another area needing further investigation is in the optimization of sample rate. The flexibility of OPDEC seems to make it a viable control technique.

# Bibliography


1. H. Kwakernaak and R. Sivan. Linear Optimal Control Systems. New York: Wiley-Interscience, 1972.

2. R. E. Kalman, P. L. Falb, and M. A. Arbib. Topics in Mathematical System Theory, McGraw-Hill, New York, 1969.

3. J. G. Reid, R. K. Mehra, and E. Kirkwood. "Robustness Properties of Output Predictive Dead-Beat Control: SISO Case", Proceedings 1979 IEEE Conference on Decision and Control, Fort Lauderdale, December 1979.

4. T. E. Fortmann and K. L. Hitz. An Introduction to Control Systems. New York: Marcel Dekker, Inc., 1977.

5. Reid, J. G. Course notes for EE 5.10, Linear Systems Theory and Digital Computation Methods. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, October 1978.

6. Houpis, Constantine H. and G. B. Lamont. Lecture Notes on Digital Control Systems/Information Processing. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, December 1977.

## Appendix A

### Condition Number Program

This Appendix contains the computer listing and a users guide for the program that finds the optimal sample rate. The optimal sample rate occurs when one over the condition number is a maximum. This program uses many of the same subroutines as the program that implements OPDEC (Appendix B). To save space, these subroutines will be discussed in Appendix B.

The user must supply the program with the following information on data cards. All data is read in using an unformated read statement.

1. System size n, n is an intiger value and must be less than or equal to ten.
2. Starting sample rate, DELT.
3. Final sample rate, DF.
4. A, B, and C from state matrix equation

   $\dot{\underline{x}} = A\underline{x} + Bu$

   $y(t) = Cx(t)$

   A, B, and C re all n by n matrices and must be discretizable over sample rate range user supplied. A must also be invertable.

Below is a brief outline of the steps the program follows:

1. Reads in data.
2. Sets DET = 0.001 and counter IS = 0.
3. Increments starting sample rate,

   DELT = DELT + DET
4. Checks to see if DELT > DF; if so, goes to step 12.

5. Discretizes A and B at sample rate DELT.

6. Increments counter IS = IS + 1.

7. Creates Hankel matrix.

8. Finds Singular values of Hankel matrix.

9. Find one over condition number,

   YS(IS) = Q(n)/Q(1)

   Q(n) is minimum singular value,
   Q(1) is maximum singular value.

10. Sets TSS(IS) = DELT.

11. Go to step 3.

12. Search reciprocal condition number array (YS) for maximum reciptrocal condition number and output this number.

13. Find the sample rate at which the maximum reciprocal condition number occurs and output this number.

14. Plot array YS vs array TSS.

15. Stop.

The program uses two subroutines from the IMSL subroutine package. The two routines are:

1. LINV2F

2. LSVDF

<u>Description of Subroutines</u>

Subroutine COND(F2,G2,C,N,IDIM,Q)

This subroutine takes the discretized system and creates the Hankel matrix from the observability and controllability pairs. Then it does a singular value decomposition of the Hankel matrix. The singular values are then output in the subscripted array Q.

Inputs

1. F2, G2, C, discretized system.
2. n, system size.
3. IDIU, Initial dimensionalization.

Outputs

1. Q, n by 1 array containing the ordered singular values

   Q(1) maximum singular value,

   Q(n) minimum singular value.

Subroutines EFT, HGRAGH, and VGRAGH are described in Appendix B.

```
      PROGRAM THESIS(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,PLOT)
      DIMENSION AT(10,10),BT(10,10),CT(10,10)
      DIMENSION Q(10)
      DIMENSION TSS(2500),YS(2500)
      DIMENSION IYSS(17)
      DIMENSION FT(10,10),GT(10,10)
      DATA IYSS(1)/20HONE OVER CONDITION  /
      DATA IYSS(3)/20H NUMBER VS SAMPLE   /
      DATA IYSS(5)/20H         RATE       /
      DATA IYSS(7)/20H                    /
      DATA IYSS(9)/20H   SAMPLE RATE      /
      DATA IYSS(11)/20HONE OVER CONDITION #/
      DATA IYSS(13)/40HONE OVER CONDITION NUMBER VS SAMPLE RATE/
      IS=0
C READING IN SIZE OF MY SYSTEM N
C N MUST BE LESS THAN OR EQUAL TO 10
      READ*,N
      IDIM=10
      IPAS=0
C READ IN INITIAL DELT
      READ*,DELT
      DET=0.001
C READ FINAL T
      READ*,DF
      READ*,((AT(I,J),J=1,N),I=1,N)
      READ*,((BT(I,J),J=1,N),I=1,N)
      READ*,((CT(I,J),J=1,N),I=1,N)
12    CONTINUE
      DELT=DET+DELT
      IF(DELT.GT.DF) GOTO 96
      IS=IS+1
      IPAS=IPAS+1
C GENERATE MY DISCRETE F AND G MATRIX FOR TRUTH AND MODEL"
      M1=55
      CALL EFT(AT,BT,N,IDIM,FT,GT,M1,DELT)
      CALL COND(FT,GT,CT,N,IDIM,Q)
      TSS(IS)=DELT
      YS(IS)=Q(N)/Q(1)
      IF(IPAS.GE.10)IPAS=0
      IF(IPAS.NE.0) GOTO 12
      PRINT*," "
      PRINT*,"  SAMPLE RATE IS  ",TSS(IS)
      PRINT*," "
      PRINT*," ONE OVER CONDITION NUMBER IS"
      PRINT*,"  1/K=",YS(IS)
      GOTO 12
96    CONTINUE
```

```
      BB=0.
      DO 197 I=1,IS,1
      IF(YS(I).GE.BB) GOTO 290
      GOTO 289
290   BB=YS(I)
      K=I
289   CONTINUE
197   CONTINUE
      PRINT*,"MAX ON OVER CONDITION #  ",BB
      PRINT*,"AT SAMPLE RATE OF  ",TSS(K)
      CALL PLOT(0.,-4.,-3)
      CALL HGRAPH(TSS,YS,IS,IYSS,1,0,1)
      CALL PLOTE(M1)
      STOP
      END .
```

```
      SUBROUTINE COND(F2,G2,C,N,IDIM,Q)
      DIMENSION F2(IDIM,IDIM),G2(IDIM,IDIM)
      DIMENSION C1(10,10),C2(10,10),C3(01,10),C4(10,10)
      DIMENSION C5(10,10),C(IDIM,IDIM)
      DIMENSION RC5(10,10)
      DIMENSION P(10,10)
      DIMENSION G7(10,10),Q(10),WK(20),B(10,10)
      CALL COPY(C,C5,N,IDIM)
      CALL COPY(G2,G7,N,IDIM)
C FIND HANKEL (IMPULSE RESPONSE) MATRIX
      DO 1115 J=1,N,1
      C2(1,J)=C(1,J)
      C4(J,1)=G7(J,1)
1115  CONTINUE
      DO 1113 I=2,N,1
      CALL MULT(C5,F2,C1,1,N,N,IDIM)
      CALL MULT(F2,G7,C3,N,N,1,IDIM)
      DO 1114 J=1,N,1
      C2(I,J)=C1(1,J)
      C4(J,I)=C3(J,1)
1114  CONTINUE
      CALL COPY(C1,C5,N,IDIM)
      CALL COPY(C3,G7,N,IDIM)
1113  CONTINUE
      CALL MULT(C2,C4,RC5,N,N,N,IDIM)
      IA=IDIM
      IDGT=N
      CALL COPY(RC5,P,N,IDIM)
      CALL LSVDF(P,IDIM,N,N,B,-1,-1,Q,WK,IER)
      RETURN
      END
```

```
C*********************************************************************
C
C
      SUBROUTINE EFT(A,B,N,IDIM,A4,B5,M,DELT)
C THIS SUBROUTINE FIND MY DISCRETE SYSTEM F AND G
C FROM A AND B.  A IS NXN AND B IS NXN
C DELT IS MY TIME INCREMENT, AND M IS THE NUMBER
C OF ITERATIONS IN MY SUM I WANT TO GO.
      DIMENSION B(IDIM,IDIM),A(IDIM,IDIM)
      DIMENSION A4(IDIM,IDIM),B5(IDIM,IDIM)
      DIMENSION A2(10,10),A3(10,10)
      DIMENSION AI(10,10)
      DIMENSION A5(10,10),AINV(10,10)
      DIMENSION B4(10,10)
      DIMENSION P(10,10)
      DIMENSION WKAREA(200)
C SET UP IDENITY MATRIX
      CALL COPY(A,P,N,IDIM)
      IA=IDIM
      IDGT=N
      DO 1002 I=1,IDIM,1
      DO 1003 J=1,N,1
      AI(I,J)=0.0
      A4(I,J)=0.0
1003  CONTINUE
      AI(I,I)=1.0
1002  CONTINUE
C FINE Q SUCH THAT F=Q+I AND G=Q*AINV*B
      C6=1.0
      DET1=1.0
      DO 1111 I=1,M,1
      DET1=DET1*DELT
      C6=C6*I
      ABLE=DET1/C6
      CALL MULT(AI,A,A2,N,N,N,IDIM)
      CALL COPY(A2,AI,N,IDIM)
      CALL MULTXK(A2,ABLE,A3,N,N,IDIM)
      CALL ADDING(A3,A4,A5,N,N,IDIM)
      CALL COPY(A5,A4,N,IDIM)
1111  CONTINUE
C FIND AINV
      CALL LINV2F(A,N,IA,AINV,IDGT,WKAREA,IER)
      CALL MULT(A4,AINV,B4,N,N,N,IDIM)
      CALL MULT(B4,B,B5,N,N,1,ICIM)
      DO 1001 I=1,N,1
      A4(I,I)=A4(I,I)+1.0
1001  CONTINUE
      CALL COPY(A,P,N,IDIM)
C A4 IS MY F AND B5 IS MY G
      RETURN
      END
C
C
C*********************************************************************
```

```
      SUBROUTINE PREDICT(F1,C,X,N,IDIM,Y)
C THIS SUBROUTINE DOES THE OUTPUT PREDICTION
C PHASE OF OPDEC.
      DIMENSION F1(IDIM,IDIM)
      DIMENSION X(IDIM,IDIM),Y(IDIM,IDIM)
      DIMENSION Z1(10,10),Z2(10,10),R(10,10)
      DIMENSION C(IDIM,IDIM)
      N4=1
      N7=N+N-1
      DO 196 II=N,N7,1
      CALL MPOWP(F1,II,N,IDIM,R)
      CALL MULT(R,X,Z1,N,N,1,IDIM)
      CALL MULT(C,Z1,Z2,1,N,1,IDIM)
      Y(N4,1)=Z2(1,1)
      N4=N4+1
196   CONTINUE
      RETURN
      END
C
C
C*****************************************************************
C
C
      SUBROUTINE ADDING(A,B,C,N,M,IDIM)
C N IS ROW, M IS COLUMN
C THIS ADDS TWO MATRICES OF SAME SIZE
      DIMENSION A(IDIM,IDIM),B(IDIM,IDIM),C(IDIM,IDIM)
      DO 906 J=1,M,1
      DO 906 I=1,N,1
      C(I,J)=A(I,J)+B(I,J)
906   CONTINUE
      RETURN
      END
C
C
C*****************************************************************
C
C
      SUBROUTINE COPY(A,B,N,IDIM)
C MUST BE A SQUARE MATRIX
C COPIES A INTO B
      DIMENSION A(IDIM,IDIM),B(IDIM,IDIM)
      DO 1100 JS=1,N,1
      DO 1100 JT=1,N,1
      B(JS,JT)=A(JS,JT)
1100  CONTINUE
      RETURN
      END
C
C
C*****************************************************************
C
C
```

```
      SUBROUTINE PRNMA(A,N,M,IDIM)
C THIS SUBROUTINE PRINTS OUT ANY SIZE MATRIX
C M IS THE NUMBER OF COLUMNS
C N IS THE NUMBER OF ROWS
      DIMENSION A(IDIM,IDIM)
      DO 1112 J=1,N,1
      WRITE(5,111)(A(J,I),I=1,M,1)
111   FORMAT(" ",10(2X,E10.4))
1112  CONTINUE
      RETURN
      END
C
C
C**************************************************************************
C
C
      SUBROUTINE MPOWP(M,NP,N,IDIM,R)
C FINDS R=M**NP
      DIMENSION M(IDIM,IDIM),R(IDIM,IDIM)
      DIMENSION R1(10,10),R2(10,10)
      DO 193 J=1,IDIM,1
      DO 194 I=1,IDIM,1
      R1(I,J)=0.0
      R2(I,J)=0.0
194   CONTINUE
      R1(J,J)=1.0
193   CONTINUE
      DO 195 JJ=1,NP,1
      CALL MULT(M,R1,R2,N,N,N,IDIM)
      CALL COPY(R2,R1,N,IDIM)
195   CONTINUE
      CALL COPY(R2,R,N,IDIM)
      RETURN
      END
C
C
C***********************************************************
C
C
      SUBROUTINE NOIZE(RMSNOTS,OUTMEAN,WN)
C****************************************
C SUBTOUTINE NOISE CALCULATES THE VALUES OF THE MEASUREMENT NOISE
C COMPONENTS USING A RANDOM NUMBER GENERATOR MODELLED AS GAUSSIAN
C****************************************
      GAUSS=0.
      DO 333 I=1,12,1
      GAUSS=GAUSS+RANF(Z10)
333   CONTINUE
      GAUSS=GAUSS-6.+OUTMEAN
      WN=GAUSS*RMSNOTS
      RETURN
      END
```

```
      SUBROUTINE HGRAPH(X,Y,N,ID,NO,NP,NS)
C IF ID(1)=0.000 BOX IN UPPER RIGHT CORNER
C IS NOT PLOTTED
      DIMENSION X(1),Y(1),ID(1) $ IF(NO.EQ.2) GO TO 30
      IF (NO.LT.0) GO TO 10
      CALL SCALE(X,7.,N,1)        $ CALL SCALE(Y,5.,N,1)
10    CALL PLOT(8.5,0.,-3) $    CALL PLOT(0.,11.,3)
      CALL PLOT(-1.35,1.35,3)
      CALL PLOT(-7.15,1.35,2)  $  CALL PLOT(-7.15,9.65,2)
      IF(ID(1).EQ.000) GO TO 25
      CALL PLOT(-7.05,9.55,3)     $ CALL PLOT(-7.05,7.55,2)
      DO 20 I=1,7,2
20    CALL SYMBOL(I*.1-6.9,7.85 ,.07,ID(I),90.,20)
      CALL PLOT(-7.05,7.55,3)     $ CALL PLOT(-6.05,7.55,2)
      CALL PLOT(-6.05,9.55,2)     $ CALL PLOT(-7.05,9.55,2)
      CALL PLOT(-7.15,9.65,3)
25    CALL PLOT(-1.35,9.65,2)    $ CALL PLOT(-1.35,1.35,2)
      CALL SYMBOL(-6.65,1.15,.1,ID(13),0.,40)
      CALL AXIS(-1.85,2.1,ID(9),-20,7.,90.,X(N+1),X(N+2))
      CALL AXIS(-1.85,2.1,ID(11),20,5.,180.,Y(N+1),Y(N+2))
 30   Y(N+2)=-Y(N+2)
      X(N+1)=X(N+1)-2.1*X(N+2)  $ Y(N+1)=Y(N+1)+1.85*Y(N+2)
      CALL LINE(Y,X,N,1,NP,NS)
      X(N+1)=X(N+1)+2.1*X(N+2)  $ Y(N+1)=Y(N+1)-1.85*Y(N+2)
      Y(N+2)=-Y(N+2)
      RETURN                     $ END
      SUBROUTINE VGRAPH(X,Y,N,ID,NO,NP,NS)
      DIMENSION X(1),Y(1),ID(1) $ IF(NO.EQ.2) GO TO 30
      IF (NO.LT.0) GO TO 10
      CALL SCALE(X,4.9,N,1)       $ CALL SCALE(Y,7.0,N,1)
10    CALL PLOT(8.5,0.,-3) $    CALL PLOT(0.,11.,3)
      CALL PLOT(-1.35,1.35,3)
      CALL PLOT(-7.15,1.35,2)  $  CALL PLOT(-7.15,9.65,2)
      CALL PLOT(-1.35,9.65,2) $ IF(ID(1).EQ.000) GO TO 25
      CALL PLOT(-1.45,9.55,3) $ CALL PLOT(-3.45,9.55,2)
      DO 20 I= 1,7,2
20    CALL SYMBOL(-3.15,9.4-I*.10,.07,ID(I),0.,20)
      CALL PLOT(-3.45,9.55,3) $ CALL PLOT(-3.45,8.35,2)
      CALL PLOT(-1.45,8.35,2) $ CALL PLOT(-1.45,9.55,2)
      CALL PLOT(-1.35,9.65,3)
25    CALL PLOT(-1.35,1.35,2)
      CALL SYMBOL(-6.65,1.15,.1,ID(13),0.,40)
      CALL AXIS(-6.4,1.85,ID(9),-20,4.9,0.,X(N+1),X(N+2))
      CALL AXIS(-6.4,1.85,ID(11),20,7.0,90.,Y(N+1),Y(N+2))
 30   X(N+1)=X(N+1)+6.4*X(N+2)    $ Y(N+1)=Y(N+1)-1.85*Y(N+2)
      CALL LINE(X,Y,N,1,NP,NS)
      X(N+1)=X(N+1)-6.4*X(N+2)    $ Y(N+1)=Y(N+1)+1.85*Y(N+2)
      RETURN      $ END
```

## Appendix B

### OPDEC Program

This appendix contains the computer listing and a users guide for the implimentation of OPDEC. The program reads in user supplied inputs and generates the initial states, $\underline{x}(0)$. The next phase is the output prediction phase. In this phase the current system state, $\underline{x}(k)$, is used to predict the outputs in the future. This predicted output is then used to calculate the next input, using OPDEC's control law. The input is then applied to the true system to update the state vector one sample step. This updated state vector is then used in the prediction phase on the next cycle through the program.

### Program Inputs and Initial Conditions

The user must supply the program with the following data. All data, unless specified otherwise, is read in using an unformated read statement.

1. A title to be used on the output plot. This title is usually the sample rate the system is using. It is used to identify output plots. The title is limited to twenty spaces and read in using an alphanumeric format.
2. The program output contains two plots. One plot is of the system output and the other is of the inputs calculated. Each plot has a title box in the upper right hand corner. This program has the option of drawing this title box or not. The next input, therefore, is a switch variable. If this variable is less than or equal to zero both title boxes will be drawn. If it is less than one but greater than zero the title box will be drawn only on the control input plot and not on the output plot. If the input is greater than one, no title box will be drawn on either plot.

3. System size n. n is an integer value and must be less than or equal to ten.

4. A, B, and C from the true system state matrix equation

$$\dot{\underline{x}} = A\,\underline{x} + B\,u$$

$$y(t) = C\,\underline{x}(t)$$

A, B, C are all read in as n by n matrices and must be discretizable at the users supplied sample rate. A must also be invertible.

5. Am, Bm, and Cm from the perturbed system state matrix equation

$$\underline{x} = Am\,\underline{x} + Bm\,u$$

$$y(t) = Cm\,\underline{x}(t)$$

Am, Bm, Cm are all read in as n by n matrices and must be discretizable at the users supplied sample rate. Am must be invertible.

6. Sample rate DELT, which the system is to use. DELT is a real number.

7. The program has an option of using a smaller model than the system order, n, in calculation of the next input. This option is the next input to the program, N6. This tells the program the size of the Hankel matrix to use in the calculation of the closed loop input. N6 is an intiger and must be less than or equal to the system size, n, and greater than or equal to 1.

8. The next input is an intiger, and tells the program the number of cycles to implement OPDEC's control law.

9. The next input is the standard deviation of the noise one wants to add to all the states.

10. RMS is the standard deviations of the guassian noise added to the input. If this value is less than or equal to zero, then no input noise is added.

11. The next inputs are three logic switches used in making the program follow a path the user desires. The inputs are read in using a logical format. The logic switches can take on two values: a T (true) or an F (false). The first logic switch, SW1, if true tells the program to use the true model in the prediction phase of the program. If SW1 is false, the program uses the perturbed model in the prediction phase of the program. The second logic switch, SW2, if true tells the program to use the true model in the control phase of the program. If SW2 is false the program uses the perturbed model in the control phase of the program. SW3 is the third logic switch. If true it will add random state noise with a standard deviation selected by the user (Input 9). If SW3 is false no state noise will be added.

After reading in the inputs, the program discretizes the true system and the perturbed system using subroutine EFT. The next sections are done sequencially in the closed loop control loop.

<u>Prediction Phase</u>

This phase uses SW1 to select which model to use in the output prediction. The output prediction is done by using subroutine PREDICT. Also done in this phase is to set up some titles to be drawn. These title changes inform the user which model was used in the prediction phase.

<u>Control Phase</u>

This phase uses SW2 to select which model to use in the control phase. The output prediction is used to determine an input. This is done by using subroutine CONTR. There are also some additional title changes, for plotting purposes, to inform the user on the output plots

which model was used in the control phase.

## System Implementation

What is done in this phase is to take the input calculated, state vector and apply them to the true system to update the state vector one sample step. This is done by using subroutine TRUTH. Also in this phase SW3 is used to add state noise if the user wants state noise added. There are some additional title changes to inform the user on the output plots if state noise, input noise or both was added.

## Output Plots

After the closed loop control is done the system then plots the sampled output versus time and control inputs versus time. The plotting is done using subroutine HGRAPH or VGRAPH.

## Major Subroutines

### Subroutine CONTR(F2,G2,C,Y,N,IDIM,D,N6,KZ,RC15)

This subroutine uses the supplied discrete system F2, G2, C and creates the Hankel matrix. The Hankel matrix is then inverted and is output in RC15. This matrix is then multiplied by the output prediction vector, Y, to determine the input D. Subroutine CONTR is called many times during the closed loop control phase. So to save computer time the Hankel matrix is created and inverted the first time CONTR is called and then stored in memory, so in subsequent callings the matrix already exists and does not need to be recomputed. N6 was described in the input section of this appendix. IDIM is the initial dimension of F2, G2, C and RC15.

### Subroutine PREDICT(F1,C,X,N,IDIM,Y)

This subroutine uses parts of the discrete system model, F1, C and state vector $\underline{x}$ to predict the output at discrete time n to 2n-1.

These outputs are then put into vector Y. IDIM is the initial dimension of Fl, C, X, and Y.

Subroutine EFT(A,B,N,IDIM,F,G,M,DELT)

This subroutine finds the discrete system F and G, from a, n by n, and B, n by n, at the sample rate of DELT.

$$F = I + \sum_{i=1}^{M} A^i \frac{(DELT)^i}{(i)!}$$

where M is the number of sumations the user wants. Then G is calculated via

$$G = (F - I) A^{-1} B$$

where I is the identity matrix. This equation is the reason why the restriction that A and Am must be invertable was stated in the input section of this appendix.

Subroutine MPOWP(M,NP,N,IDIM,R)

This subroutine takes matrix M, n by n, to the power of NP. Np is an integer value. The answer is then put into matrix R. IDIM is the initial dimension of M and R.

Subroutine TRUTH(A,B,C,DELT,D,X,IDIM,N,TSS,YS,IS,US,RMS)

This subroutine takes the input D and state vector $\underline{x}$ and applies them to the true system, A, B, C to update the state vector one sample rate. To insure the complete output response and help smooth output plots the true system runs at a sample rate that is eleven times faster than the user's selected sample rate. First the true system is discretized using subroutine EFT, at a sample rate of DELL, where

$$DELL = DELT/11.0$$

Next, input noise is added if RMS is greater than zero. Then a white

guassian noise will be added to the input. The noise will have a standaru deviation of RMS. Next the system states are updated one sample rate. Also the arrays to be plotted TSS, US, and YS are created. IS is a counter for setting up these arrays. Then after the state vector has been updated the time, state vector and output are printed.

Subroutine NOIZE(RMSNOTS,OUTMEAN,WN)

This subroutine uses a random number generator to create a white guassian noise with a standard deviation of RMSNOTS and a mean of OUTMEAN. The noise is output in WN.

Closing Remarks

This program can only be used with a single input, single output system. But all of the subroutines and the vital parts in the main program are set up for a system with n inputs and n outputs. The plotting routines are set up so that with minor modifications one can have multi-plots per run. The program uses some subroutines from IMSL library packages, they are LINV2F and LSVDF.

```
      PROGRAM THESIS(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,PLOT)
      DIMENSION AT(10,10),BT(10,10),CT(10,10)
      DIMENSION AM(10,10),BM(10,10),CM(10,10)
      DIMENSION X0(10,10),Y(10,10)
      DIMENSION X(10,10),RC15(10,10)
C MAX NUMBER OF INPUTS ONE CAN CALCULATE WITH DIMENSION OF 1000
C IS 56 BECAUSE OF DIVISION BY 11 IN TRUTH SUBROUTINE
      DIMENSION TSS(1000),YS(1000),US(1000)
      DIMENSION IYSS(17),IDSS(17),IZSS(17),IXSS(17),ISSS(17)
      DIMENSION CON1(10,10),CON2(10,10)
      DIMENSION FT(10,10),FM(10,10),GT(10,10),GM(10,10)
      COMMON/MAIN1/NDIM,NDIM1,CON1/MAIN2/CON2/INOU/KIN,KOUT,KPUNCH
      LOGICAL SW1,SW2,SW3
C SET UP TITLES FOR CALCOMP PLOTS
      DATA IYSS(1)/20H  SAMPLED OUTPUT    /
      DATA IYSS(9)/20H  TIME IN SECONDS   /
      DATA IYSS(11)/20H  SAMPLED OUTPUT    /
      DATA IDSS(1)/20H  CONTROL INPUTS    /
      DATA IDSS(9)/20H  TIME IN SECONDS   /
      DATA IDSS(11)/20H  CONTROL INPUTS    /
      DATA IZSS(1)/20H  TRUE PREDICTION   /
      DATA IZSS(3)/20H  FALSE PREDICTION  /
      DATA IZSS(5)/20HTRUE CONTROL MODEL  /
      DATA IZSS(7)/20HFALSE CONTROL MODEL /
      DATA IZSS(9)/40HSAMPLED OUTPUT WITH STATE NOISE ADDED   /
      DATA IZSS(13)/40HSAMPLED OUTPUT WITH NO NOISE ADDED      /
      DATA IXSS(1)/40HCONTROL INPUTS WITH STATE NOISE ADDED   /
      DATA IXSS(5)/40HCONTROL INPUTS WITH NO NOISE ADDED      /
      DATA IXSS(9)/40HOUTPUT WITH INPUT AND STATE NOISE ADDED /
      DATA IXSS(13)/40HCONTROL INPUT WITH INPUT AND STATE NOISE/
      DATA ISSS(1)/40HSAMPLED OUTPUT WITH INPUT NOISE ADDED   /
      DATA ISSS(5)/40HCONTROL INPUT WITH INPUT NOISE ADDED    /
C READING IN 4TH LINE OF UPPER RIGHT HAND BOX OF PLOTS
C USUALLY READ IN IS "SAMPLE RATE OF XXXXX"
      READ(5,556)IYSS(3),IYSS(4)
556   FORMAT(1X,2A10)
      IDSS(3)=IYSS(3)
      IDSS(4)=IYSS(4)
C READ A NUMBER TO MAKE BOX IN UPPER LEFT HAND CORNER
C OR NOT I ZZZ.GT.1.0 THEN NO BOX IN CONTROL
C IF ZZZ.GT.0. BUT.LE.1.0 THEN NO BOX IN OUTPUT PLOT
C BUT BOX IN CONTROL PLOT
      READ*,ZZZ
      IF(ZZZ.GT.0.0)IYSS(1)=0.0
      IF(ZZZ.GT.1.0)IDSS(1)=0.0
C READING IN SIZE OF MY SYSTEM N
C N MUST BE LESS THAN OR EQUAL TO 10
      READ*,N
      IDIM=10
      Z9=55
      NDIM=10
      NDIM1=11
```

```
      KIN=5
      KOUT=5
      KPUNCH=7
C READING IN TRUE SYSTEM IN STATE MATRIX FORM
      READ*,((AT(I,J),J=1,N),I=1,N)
      READ*,((BT(I,J),J=1,N),I=1,N)
      READ*,((CT(I,J),J=1,N),I=1,N)
C READ IN PERTURBED SYSTEM IN STATE MATRIX FORM
      READ*,((AM(I,J),J=1,N),I=1,N)
      READ*,((BM(I,J),J=1,N),I=1,N)
      READ*,((CM(I,J),J=1,N),I=1,N)
12    CONTINUE
C READ IN SAMPLE RATE USED
      READ*,DELT
C TO STOP PROGRAM LET DELT BE .LE.0.0
      IF(DELT.LE.0.0) GOTO 96
C READING SIZE OF HANKEL MATRIX USED IN INVERSE
C IN SUBROUTINE CONTR
      READ*,NS
C READING IN THE NUMBER OF TIMES YOU CALCULATE AN INPUT
      READ*,KZ
      PRINT*,"DELT=",DELT
C READING RMS VALUES FOR NOISES
C FIRST IS FOR NOISE ADDED TO STATES
C THEN NOISE ADDED TO INPUTS
C IF RMS LESS THAN ZERO NO NOISE ADDED TO INPUTS
      READ*,RMSNOIS,RMS
C READ IN LOGIC SWITH FOR COMPUTATION
C SW1 FOR PREDICTION SW1=TRUE USING MATRIX AT FOR PREDICTION
C SW1=FALSE, USING MATRIX AM FOR PREDICTION
C SW2=TRUE, USING TRUE MODEL FOR CONTROL PART OF PROBLEM
C SW2=FALSE, USING PERTURBED MODEL FOR CONTROL PART OF PROBLEM
C SW3=TRUE, USING NOISE IN STATE UPDATE
C SW3=FALSE, NO NOISE BEING USED.
      READ(5,555) SW1,SW2,SW3
555   FORMAT(3L1)
C SET COUNTER TO ZERO
      IS=0
C GENERATE MY DISCRETE F AND G MATRIX FOR TRUTH AND MODEL SYSTEM
      M1=55
      CALL EFT(AT,BT,N,IDIM,FT,GT,M1,DELT)
      CALL EFT(AM,BM,N,IDIM,FM,GM,M1,DELT)
C GENERATE INITIAL CONDITIONS RANDOMLY
      CALL RANSET(Z9)
      DO 444 I=1,N,1
      X0(I,1)=10*RANF(Z9)
444   CONTINUE
      PRINT*," "
      PRINT*," INITIAL CONDITIONS ARE"
      CALL PRNMA(X0,N,1,IDIM)
      CALL COPY(X0,X,N,IDIM)
      PRINT*," "
```

```
      DO 445 KNI=1,KZ,1
C GOTO PROPER PREDICTION USING LOGIC SWITCH #1
      IF(SW1)GOTO 29
      IYSS(5)=IZSS(3)
      IYSS(5)=IZSS(4)
      IDSS(5)=IZSS(3)
      IDSS(6)=IZSS(4)
      PRINT*,"USING MODEL MATRIX SW1 IS FALSE"
      CALL PREDICT(FM,CM,X,N,IDIM,Y)
      GOTO 30
29    PRINT*,"USING TRUTH MATRIX SW1 IS TRUE"
      IYSS(5)=IZSS(1)
      IYSS(6)=IZSS(2)
      IDSS(5)=IZSS(1)
      IDSS(6)=IZSS(2)
      CALL PREDICT(FT,CT,X,N,IDIM,Y)
30    PRINT*," MY PREDICTION IS "
      CALL PRNMA(Y,N,1,IDIM)
      PRINT*,"  "
C GOTO PROPER CONTROL USING LOGIC SWITCH #2
      IF(SW2) GOTO 39
      IYSS(7)=IZSS(7)
      IYSS(8)=IZSS(8)
      IDSS(7)=IZSS(7)
      IDSS(8)=IZSS(8)
      PRINT*,"USING MODEL MATRIX FOR CONTROL SW2 IS FALSE"
      CALL CONTR(FM,GM,CM,Y,N,IDIM,D,N6,KNI,RC15)
      GOTO 40
39    PRINT*,"USING TRUTH MATRIX FOR CONTROL SW2 IS TRUE"
      IYSS(7)=IZSS(5)
      IYSS(8)=IZSS(6)
      IDSS(7)=IZSS(5)
      IDSS(8)=IZSS(6)
      CALL CONTR(FT,GT,CT,Y,N,IDIM,D,N6,KNI,RC15)
40    PRINT*,"  "
      PRINT*,"MY INPUT IS  ",D
      PRINT*,"  "
      CALL TRUTH(AT,BT,CT,DELT,D,X0,IDIM,N,TSS,YS,IS,US,RMS)
C PUT IN NOISE USING SWITCH #3
      IF(RMS.LE.0.) GOTO 798
      IF(SW3) GOTO 49
      PRINT*," "
      PRINT*,"INPUT NOISE ADDED BUT NO STATE NOISE ADDED"
      DO 866 KN=13,16,1
      IYSS(KN)=ISSS(KN-12)
      IDSS(KN)=ISSS(KN-8)
866   CONTINUE
      CALL COPY(X0,X,N,IDIM)
      GOTO 445
49    PRINT*,"  "
      PRINT*,"BOTH INPUT NOISE AND STATE NOISE ADDED"
      DO 867 KN=13,16,1
      IYSS(KN)=IXSS(KN-4)
```

```
      IDSS(KN)=IXSS(KN)
867   CONTINUE
      GOTO 778
798   IF(SW3) GOTO 799
      PRINT*,"NO INPUT NOISE OR STATE NOISE ADDED"
      DO 878 KN=13,16,1
      IYSS(KN)=IZSS(KN)
      IDSS(KN)=IXSS(KN-8)
878   CONTINUE
      CALL COPY(X0,X,N,IDIM)
      GOTO 445
799   PRINT*,"NO INPUT NOISE BUT STATE NOISE ADDED"
      DO 879 KN=13,16,1
      IYSS(KN)=IZSS(KN-4)
      IDSS(KN)=IXSS(KN-12)
879   CONTINUE
778   CONTINUE
      DO 446 KP=1,N,1
      OUTMEAN=0.0
      CALL NOIZE(RMSNOIS,OUTMEAN,WN)
      X(KP,1)=X0(KP,1)+WN
446   CONTINUE
      PRINT*,"STATES AFTER NOISE ADDED"
      CALL PRNMA(X,N,1,IDIM)
445   PRINT*,"  "
      CALL PLOT(0.,-4.,-3)
      CALL HGRAPH(TSS,YS,IS,IYSS,1,0,1)
      CALL PLOT(0.,-4.,-3)
      CALL HGRAPH(TSS,US,IS,IDSS,1,0,1)
      GOTO 12
96    CONTINUE
      CALL PLOTE(M1)
      STOP
      END
```

```
C
C
C********************************************************************
      SUBROUTINE CONTR(F2,G2,C,Y,N,IDIM,D,N6,KZ,RC15)
C THIS SUBROUTINE CREATES THE HANKEL MATRIX
C AND USES IT FOR COMPUTING THE INPUT NEEDED
      DIMENSION F2(IDIM,IDIM),G2(IDIM,IDIM)
      DIMENSION C1(10,10),C2(10,10),C3(01,10),C4(10,10)
      DIMENSION C5(10,10),C(IDIM,IDIM)
      DIMENSION RC5(10,10),RC15(IDIM,IDIM)
      DIMENSION P(10,10)
      DIMENSION G7(10,10),Q(10),WK(20),B(10,10)
      DIMENSION Y(IDIM,IDIM),U(10,10)
      IF(KZ.GT.1) GOTO 666
      CALL COPY(C,C5,N,IDIM)
      CALL COPY(G2,G7,N,IDIM)
C FIND HANKEL (IMPULSE RESPONSE) MATRIX
      DO 1115 J=1,N,1
      C2(1,J)=C(1,J)
      C4(J,1)=G7(J,1)
1115  CONTINUE
      DO 1113 I=2,N,1
      CALL MULT(C5,F2,C1,1,N,N,IDIM)
      CALL MULT(F2,G7,C3,N,N,1,IDIM)
      DO 1114 J=1,N,1
      C2(I,J)=C1(1,J)
      C4(J,I)=C3(J,1)
1114  CONTINUE
      CALL COPY(C1,C5,N,IDIM)
      CALL COPY(C3,G7,N,IDIM)
1113  CONTINUE
      CALL MULT(C2,C4,RC5,N,N,N,IDIM)
      IA=IDIM
      IDGT=N
      PRINT*,"HANKEL MATRIX IS"
      CALL PRNMA(RC5,N6,N6,IDIM)
      CALL COPY(RC5,P,N,IDIM)
      CALL LSVDF(P,IDIM,N,N,B,-1,-1,Q,WK,IER)
      PRINT*,"  "
      PRINT*," SINGULAR VALUES OF HANKEL MATRIX"
      CALL PRNMA(Q,N,1,IDIM)
      CALL GMINV(N6,N6,RC5,RC15,MR,5)
      PRINT*,"RANK OF HANKEL MATRIX IS  ",MR
666   CONTINUE
      CALL MULT(RC15,Y,U,N6,N6,1,IDIM)
      D=-U(N6,1)
      RETURN
      END
C
C
C********************************************************************
C
C
```

```
      SUBROUTINE PREDICT(F1,C,X,N,IDIM,Y)
C THIS SUBROUTINE DOES THE OUTPUT PREDICTION
C PHASE OF OPDEC.
      DIMENSION F1(IDIM,IDIM)
      DIMENSION X(IDIM,IDIM),Y(IDIM,IDIM)
      DIMENSION Z1(10,10),Z2(10,10),R(10,10)
      DIMENSION C(IDIM,IDIM)
      N4=1
      N7=N+N-1
      DO 196 II=N,N7,1
      CALL MPOWP(F1,II,N,IDIM,R)
      CALL MULT(R,X,Z1,N,N,1,IDIM)
      CALL MULT(C,Z1,Z2,1,N,1,IDIM)
      Y(N4,1)=Z2(1,1)
      N4=N4+1
196   CONTINUE
      RETURN
      END
C
C
C*****************************************************************
C
C
      SUBROUTINE ADDING(A,B,C,N,M,IDIM)
C N IS ROW, M IS COLUMN
C THIS ADDS TWO MATRICES OF SAME SIZE
      DIMENSION A(IDIM,IDIM),B(IDIM,IDIM),C(IDIM,IDIM)
      DO 906 J=1,M,1
      DO 906 I=1,N,1
      C(I,J)=A(I,J)+B(I,J)
906   CONTINUE
      RETURN
      END
C
C
C*****************************************************************
C
C
      SUBROUTINE COPY(A,B,N,IDIM)
C MUST BE A SQUARE MATRIX
C COPIES A INTO B
      DIMENSION A(IDIM,IDIM),B(IDIM,IDIM)
      DO 1100 JS=1,N,1
      DO 1100 JT=1,N,1
      B(JS,JT)=A(JS,JT)
1100  CONTINUE
      RETURN
      END
C
C
C*****************************************************************
C
C
```

```
C*************************************************************
C
C
      SUBROUTINE EFT(A,B,N,IDIM,A4,B5,M,DELT)
C THIS SUBROUTINE FIND MY DISCRETE SYSTEM F AND G
C FROM A AND B.  A IS NXN AND B IS NXN
C DELT IS MY TIME INCREMENT, AND M IS THE NUMBER
C OF ITERATIONS IN MY SUM I WANT TO GO.
      DIMENSION B(IDIM,IDIM),A(IDIM,IDIM)
      DIMENSION A4(IDIM,IDIM),B5(IDIM,IDIM)
      DIMENSION A2(10,10),A3(10,10)
      DIMENSION AI(10,10)
      DIMENSION A5(10,10),AINV(10,10)
      DIMENSION B4(10,10)
      DIMENSION P(10,10)
      DIMENSION WKAREA(200)
C SET UP IDENITY MATRIX
      CALL COPY(A,P,N,IDIM)
      IA=IDIM
      IDGT=N
      DO 1002 I=1,IDIM,1
      DO 1003 J=1,N,1
      AI(I,J)=0.0
      A4(I,J)=0.0
1003  CONTINUE
      AI(I,I)=1.0
1002  CONTINUE
C FINE Q SUCH THAT F=Q+I AND G=Q*AINV*B
      C6=1.0
      DET1=1.0
      DO 1111 I=1,M,1
      DET1=DET1*DELT
      C6=C6*I
      ABLE=DET1/C6
      CALL MULT(AI,A,A2,N,N,N,IDIM)
      CALL COPY(A2,AI,N,IDIM)
      CALL MULTXK(A2,ABLE,A3,N,N,IDIM)
      CALL ADDING(A3,A4,A5,N,N,IDIM)
      CALL COPY(A5,A4,N,IDIM)
1111  CONTINUE
C FIND AINV
      CALL LINV2F(A,N,IA,AINV,IDGT,WKAREA,IER)
      CALL MULT(A4,AINV,B4,N,N,N,IDIM)
      CALL MULT(B4,B,B5,N,N,1,IDIM)
      DO 1001 I=1,N,1
      A4(I,I)=A4(I,I)+1.0
1001  CONTINUE
      CALL COPY(A,P,N,IDIM)
C A4 IS MY F AND B5 IS MY G
      RETURN
      END
C
C
C*************************************************************
```

```
C
C
C************************************************************
C
C
      SUBROUTINE TRANSP(A,B,N,IDIM)
C THIS TRANSPOSES A AND PUTS INTO B
      DIMENSION A(IDIM,IDIM)
      DIMENSION B(IDIM,IDIM)
      DO 1300 J=1,N,1
      DO 1300 I=1,N,1
      B(I,J)=A(J,I)
1300  CONTINUE
      RETURN
      END
C
C
C************************************************************
C
C
      SUBROUTINE MULTXK(A,D,C,M,N,IDIM)
C THIS MULTPLIES A MATRIX BY A CONSTANT
      DIMENSION A(IDIM,IDIM),C(IDIM,IDIM)
      DO 905 J=1,M,1
      DO 905 I=1,N,1
      C(J,I)=A(J,I)*D
905   CONTINUE
      RETURN
      END
C
C
C************************************************************
C
C
      SUBROUTINE MULT(S,ST,H,L,M,N,IDIM)
C THIS MULTPLIES TWO MATRICIES TOGETHER
      DIMENSION S(IDIM,IDIM),ST(IDIM,IDIM),H(IDIM,IDIM)
      DO 2000 I=1,L,1
      DO 2000 K=1,N,1
      SUM=0.0
      DO 2001 J=1,M,1
      SUM=SUM+S(I,J)*ST(J,K)
2001  CONTINUE
      H(I,K)=SUM
2000  CONTINUE
      RETURN
      END
C
C
C************************************************************
C
C
```

```
C
C
C*****************************************************************
C
C
      SUBROUTINE TRUTH(A1,B1,C8,DELT,D,X,IDIM,N,TSS,YS,IS,US,RMS)
C THIS IS THE TRUE SYSTEM RESPONSE WHICH IS RUNNING
C ELEVEN TIMES FASTR, SO THAT THE OUTPUT PLOTS ARE SMOOTH
C IT ALSO CREATES THE ARRAYS FOR CALCOMP PLOTS
      DIMENSION X(IDIM,IDIM),TSS(1000)
      DIMENSION A1(10,10),B1(10,10),C8(10,10)
      DIMENSION Z6(10,10),Z7(10,10)
      DIMENSION F3(10,10),G3(10,10)
      DIMENSION SY(10,10),YS(1000),US(1000)
      M=50
      DELL=DELT/11.0
      CALL EFT(A1,B1,N,IDIM,F3,G3,M,DELL)
      DO 702 IP=1,11,1
      IS=IS+1
      IF(IS.EQ.1) GOTO 44
      TSS(IS)=TSS(IS-1)+DELL
      GOTO 45
44    TSS(IS)=DELL
45    CONTINUE
      B=D
      IF(RMS.LE.0.0) GOTO 777
      OUTMEAN=0.
      CALL NOIZE(RMS,OUTMEAN,WN)
      B=D+WN
777   CONTINUE
      US(IS)=B
776   CONTINUE
      CALL MULT(F3,X,Z6,N,N,1,IDIM)
      CALL MULTXK(G3,B,Z7,N,1,IDIM)
      CALL ADDING(Z6,Z7,X,N,1,IDIM)
      CALL MULT(C8,X,SY,1,N,1,IDIM)
      YS(IS)=SY(1,1)
702   CONTINUE
      PRINT*," TIME IS  ",TSS(IS),"  MY STATES ARE "
      CALL PRNMA(X,N,1,IDIM)
      PRINT*,"  "
      PRINT*," OUTPUT IS  ",YS(IS)
      PRINT*," "
      RETURN
      END
C
C
C*****************************************************************
C
C
```

```
      SUBROUTINE PRNMA(A,N,M,IDIM)
C THIS SUBROUTINE PRINTS OUT ANY SIZE MATRIX
C M IS THE NUMBER OF COLUMNS
C N IS THE NUMBER OF ROWS
      DIMENSION A(IDIM,IDIM)
      DO 1112 J=1,N,1
      WRITE(5,111)(A(J,I),I=1,M,1)
111   FORMAT(" ",10(2X,E10.4))
1112  CONTINUE
      RETURN
      END
C
C
C*****************************************************************************
C
C
      SUBROUTINE MPOWP(M,NP,N,IDIM,R)
C FINDS R=M**NP
      DIMENSION M(IDIM,IDIM),R(IDIM,IDIM)
      DIMENSION R1(10,10),R2(10,10)
      DO 193 J=1,IDIM,1
      DO 194 I=1,IDIM,1
      R1(I,J)=0.0
      R2(I,J)=0.0
194   CONTINUE
      R1(J,J)=1.0
193   CONTINUE
      DO 195 JJ=1,NP,1
      CALL MULT(M,R1,R2,N,N,N,IDIM)
      CALL COPY(R2,R1,N,IDIM)
195   CONTINUE
      CALL COPY(R2,R,N,IDIM)
      RETURN
      END
C
C
C*********************************************************************
C
C
      SUBROUTINE NOIZE(RMSNOTS,OUTMEAN,WN)
C**************************************************
C SUBTOUTINE NOISE CALCULATES THE VALUES OF THE MEASUREMENT NOISE
C COMPONENTS USING A RANDOM NUMBER GENERATOR MODELLED AS GAUSSIAN
C**************************************************
      GAUSS=0.
      DO 333 I=1,12,1
      GAUSS=GAUSS+RANF(Z10)
333   CONTINUE
      GAUSS=GAUSS-6.+OUTMEAN
      WN=GAUSS*RMSNOTS
      RETURN
      END
```

```
      SUBROUTINE HGRAPH(X,Y,N,ID,NO,NP,NS)
C IF ID(1)=0.000 BOX IN UPPER RIGHT CORNER
C IS NOT PLOTTED
      DIMENSION X(1),Y(1),ID(1) $ IF(NO.EQ.2) GO TO 30
      IF (NO.LT.0) GO TO 10
      CALL SCALE(X,7.,N,1)        $ CALL SCALE(Y,5.,N,1)
10    CALL PLOT(8.5,0.,-3) $    CALL PLOT(0.,11.,3)
      CALL PLOT(-1.35,1.35,3)
      CALL PLOT(-7.15,1.35,2)  $  CALL PLOT(-7.15,9.65,2)
      IF(ID(1).EQ.000) GO TO 25
      CALL PLOT(-7.05,9.55,3)      $ CALL PLOT(-7.05,7.55,2)
      DO 20 I=1,7,2
20    CALL SYMBOL(I*.1-6.9,7.65 ,.07,ID(I),90.,20)
      CALL PLOT(-7.05,7.55,3)      $ CALL PLOT(-6.05,7.55,2)
      CALL PLOT(-6.05,9.55,2)      $ CALL PLOT(-7.05,9.55,2)
      CALL PLOT(-7.15,9.65,3)
25    CALL PLOT(-1.35,9.65,2)    $ CALL PLOT(-1.35,1.35,2)
      CALL SYMBOL(-6.65,1.15,.1,ID(13),0.,40)
      CALL AXIS(-1.85,2.1,ID(9),-20,7.,90.,X(N+1),X(N+2))
      CALL AXIS(-1.85,2.1,ID(11),20,5.,180.,Y(N+1),Y(N+2))
30    Y(N+2)=-Y(N+2)
      X(N+1)=X(N+1)-2.1*X(N+2)  $ Y(N+1)=Y(N+1)+1.85*Y(N+2)
      CALL LINE(Y,X,N,1,NP,NS)
      X(N+1)=X(N+1)+2.1*X(N+2)  $ Y(N+1)=Y(N+1)-1.85*Y(N+2)
      Y(N+2)=-Y(N+2)
      RETURN                      $ END
      SUBROUTINE VGRAPH(X,Y,N,ID,NO,NP,NS)
      DIMENSION X(1),Y(1),ID(1) $ IF(NO.EQ.2) GO TO 30
      IF (NO.LT.0) GO TO 10
      CALL SCALE(X,4.9,N,1)        $ CALL SCALE(Y,7.0,N,1)
10    CALL PLOT(8.5,0.,-3) $    CALL PLOT(0.,11.,3)
      CALL PLOT(-1.35,1.35,3)
      CALL PLOT(-7.15,1.35,2)  $  CALL PLOT(-7.15,9.65,2)
      CALL PLOT(-1.35,9.65,2) $ IF(ID(1).EQ.000) GO TO 25
      CALL PLOT(-1.45,9.55,3) $ CALL PLOT(-3.45,9.55,2)
      DO 20 I= 1,7,2
20    CALL SYMBOL(-3.15,9.4-I*.10,.07,ID(I),0.,20)
      CALL PLOT(-3.45,9.55,3) $ CALL PLOT(-3.45,8.55,2)
      CALL PLOT(-1.45,8.55,2) $ CALL PLOT(-1.45,9.55,2)
      CALL PLOT(-1.35,9.65,3)
25    CALL PLOT(-1.35,1.35,2)
      CALL SYMBOL(-6.65,1.15,.1,ID(13),0.,40)
      CALL AXIS(-6.4,1.85,ID(9),-20,4.9,0.,X(N+1),X(N+2))
      CALL AXIS(-6.4,1.85,ID(11),20,7.0,90.,Y(N+1),Y(N+2))
30    X(N+1)=X(N+1)+6.4*X(N+2)   $ Y(N+1)=Y(N+1)-1.85*Y(N+2)
      CALL LINE(X,Y,N,1,NP,NS)
      X(N+1)=X(N+1)-6.4*X(N+2)   $ Y(N+1)=Y(N+1)+1.85*Y(N+2)
      RETURN      $ END
```

# Vita

Earl H. Kirkwood Jr. was born on 5 March 1956, in Saint Louis, Missouri. He was graduated from the Lafayette Senior High School in Ellisville, Missouri, in 1974. He then entered the University of Missouri in Columbia, Missouri were he studied Electrical Engineering. On May 13, 1978 he was awarded a Bachelor of Science Degree in Electrical Engineering with Honors. Upon graduation he received a commission into the United States Air Force and was selected to join the resident graduate Guidance and Control program under the department of Aeronautics and Astronautics at the U.S. Air Force Institute of Technology.

Permanent Address: 1016 Mallow Drive
Ellisville, Missouri 63011

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|---|
| 1. REPORT NUMBER AFIT/GA/EE/80-1 | 2. GOVT ACCESSION NO. AD-A085 709 | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle) ROBUSTNESS STUDIES OF OUTPUT PREDICTIVE DEAD BEAT CONTROLL (OPDEC) FOR WING FLUTTER APPLICATION | | 5. TYPE OF REPORT & PERIOD COVERED MASTER THESIS |
| | | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s) EARL H. KIRKWOOD JR. 2nd Lt. USAF | | 8. CONTRACT OR GRANT NUMBER(s) |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT-EN) Wright-Patterson AFB, OHIO 45433 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS |
| 11. CONTROLLING OFFICE NAME AND ADDRESS | | 12. REPORT DATE December 1979 |
| | | 13. NUMBER OF PAGES 134 |
| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Air Force Flight Dynamic Laboratory (AFFDL-FGC) Air Force Systems Command Wright-Patterson AFB, OHIO 45433 | | 15. SECURITY CLASS. (of this report) Unclassified |
| | | 15a. DECLASSIFICATION DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for Public Release; IAW AFR 190-17

J. P. Hipps, Major, USAF
Director of Public Affairs

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

OUTPUT PREDICTIVE DEAD BEAT CONTROL
HANKEL MATRIX
IMPULSE RESPONSE
ROBUSTNESS
SINGULAR VALUES

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A new discrete control law is implemented and examined. Robustness to noise and model errors of the control law as sample rate varies is analyzed. This analysis is conducted while controlling several different very lightly damped, single input/single output systems which are representative of the flutter dynamics of the B-52E wing.

The control law performance is different at separate sample rates. A distinct range of sample rates are found to have a better response to noise than other sample rates. Another range is found to be more robust when there

exists an error in the models used to calculate the closed loop control law. When these ranges of sample rates intersect, the robustness characteristics at those sample rates is found to be good with respect to both noise and model mismatch.

As theoretically predicted a relationship between condition number of the system Hankel matrix and robustness seems to exist. Hence, these simulated results appear to validate the theoretical results on robustness predicted by Reid, but on the other hand, these simulated results indicate that the total analysis of "robustness" is a very conplex issue and cannot, at this point, be totally predicted by such a parameter as simple as the condition number of the Hankel matrix.